

BÖLÜM

7

11. Yazılım Gerçekleştirimi

8. Hafta

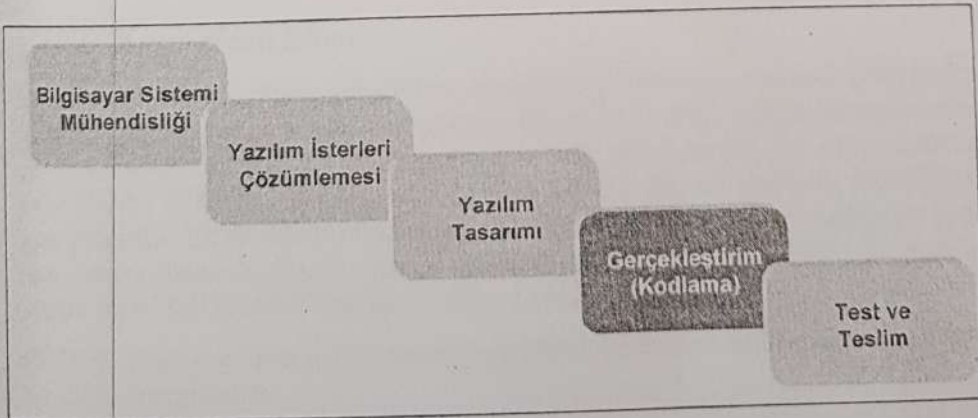
1. Grup

Boşluk

Yazılım mühendisliği ana aşamalarından çözümleme ve tasarım, aslında isterlerin bilgisayarın anlayabileceği şekle dönüştürülmesi için bir altyapı sağlar. Çözümleme ve tasarım, düşüncelerin makinelerin anlayabileceği hale getirilmesine yardımcı olurken, *gerçekleştirim* aşamasında, tasarım, programlama diline dönüştürülür. Bilgisayara yaptırılması istenen görevleri ona tanıtabilmek için programlama dilleri kullanılır; yazılan kaynak kod daha sonra bilgisayar mimarisine uygun makine kodlarına dönüştürülerek uygun bir merkezi işlem birimi üzerinde çalıştırılır. Gerçekleştirim aşaması sonunda, yürütülebilir kod veya dinamik kütüphane şeklinde yazılım birimleri elde edilir. Pek çok yerde bu birimlere *program* denmektedir.

Bu bölümde, gerçekleştirim, yani kodlama aşamasının genel özellikleri üzerinde durulacak, kodlama biçimlerine değinilecek, fakat belirli bir dilde nasıl kod yazılacağı ve ne şekilde derleme ve bağlama yapılacağı anlatılmayacaktır.

Şekil-7.1 de yeri gösterilen gerçekleştirim aşamasında, tasarım, bir geliştirme ortamı üzerinde önce programlama diline ve sonra da makine koduna dönüştürülerek yürütülebilir kod (executable) oluşturulur.



Şekil-7.1. Yazılım gerçekleştiriminin yeri.

1.1. Programlama Dilleri

Yazılım mühendisliğinin diğer evreleri ne kadar nitelikli olursa olsun, gerçekleştirim, yani kodlama evresinde kullanılan programlama dili ve kodlama biçimi son ürünün hem niteliğini hem de bakım özelliğini etkiler. Ayrıca kodlayıcıların verimliliğini de etkileyen programlama dili seçimi genel başarıyı artırabilmek için özel bir önem gerektirir. Seçilen programlama dili, geliştirme ortamı ve kodlayıcı personelin deneyimi, süreç sonunda ortaya çıkan yazılım ürününün genel niteliğini çok büyük ölçüde etkiler. Dolayısıyla, programlama dillerinin tarihçesine kısaca değinmekte yarar vardır [5] [11].

1.1.1. Dillerin Tarihçesi

1960'larda yapısal programlama yöntemi ortaya çıktı. Bu yöntemi destekleyen programlama dilleri ile nispeten karmaşık ve büyük programlar yazabilmek mümkün oluyordu. Fakat yine de projeler belirli boyutları aştığında yapısal programlama yöntemi da yetersiz kalmakta ve projelerin denetimi zorlaşmaktaydı. Yapısal programlama teknikleri ile yapılmış birçok proje artık bu yöntemin yetersiz kaldığı bir noktaya gelmişti.

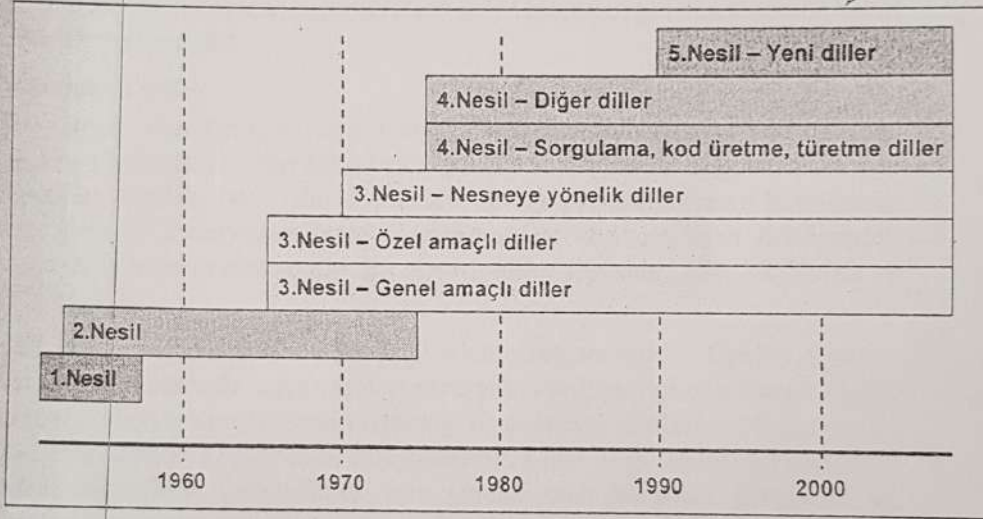
Programcılığa yaklaşımlar bilgisayarın icadından bu yana pek çok kereler değişikliğe uğramıştır. Bunun en önemli nedenlerinden biri gittikçe karmaşıklaşan program isteklerine cevap verebilmektir. İlk bilgisayar programları *Assembly* dili ile yazılmaktaydı. Program boyutları büyüdükçe daha da üst düzey diller ortaya çıktı. Yaygın olarak kullanılan ilk üst düzey programlama dili FORTRAN'dır (Formula Translator). Ancak çeşitli eksiklik ve kısıtlamalar nedeniyle FORTRAN ile açık, anlaşılması kolay, uzun ve kapsamlı programlar yazmak zordu.

Çeşitli gereksinimler nedeniyle bilgisayar programlama tekniklerine 1980'lerde yeni bir yaklaşım daha eklenmiştir. Günümüzde çok yaygın olan bu yazılım geliştirme tekniğine *Nesneye Yönelik Çözümleme ve Tasarım* (Object-Oriented Analysis and Design) ve bunun sonucunda ortaya çıkan programlama yöntemine de *Nesneye Yönelik Programlama* (Object Oriented Programming) adı verilmiştir. Bu yöntem başarılı bir proje yaratabilmek için kodlayıcıya yapısal programlamanın yanında yeni bir yol göstermektedir.

1980'lerin yaygın yüksek düzey dillerinden biri olan C programlama dili, etkinliği, programlama kuvveti ve kullanım kolaylıkları ile tanınmaktadır. C++ ise C'nin nesneye yönelik uyarlamasıdır. C'nin yeteneklerini taşıması yanında nesneye yönelimi de desteklemesi bu dili benzerleri içinde en yaygın ve önemli hale getirmiştir. Ancak, C++'ı yalnızca yapısal bir dil olarak kullanmaya çalışmanın otoyolda birinci vitesle araç kullanmaya benzediği de kabul edilmelidir. Günümüzde de çok yeni sayılabilecek, fakat çok yaygın bir dil olan JAVA daha çok İnternet uygulamaları ve taşınabilir programlama için ideal olarak görülmektedir. Bunların yanında taşınabilirliği ve uyumluluğu sağlamak üzere HTML, XML gibi yapısal diller ve çeşitli özel amaçlı toplu iş tanımlama dilleri de yaygın bir şekilde kullanılmaktadır.

11.1.2. Dillerin Gelişimi

Bilgisayarın icadından bu yana çeşitli yazılım geliştirme projelerinde kullanılmış olan yüzlerce programlama dili bulunmaktadır. Tarihçesine göre bu diller beş ana nesil içinde toplanabilirler. Şekil-7.2 de bir zaman çizgisi üzerinde gösterilen dillerin gelişimini şimdi kısaca inceleyelim:



Şekil-7.2. Programlama dillerinin gelişimi.

11.1.2.1. Birinci Nesil Diller

İlk ortaya çıkan programlama dilleri makine düzeyinde kodlamaya, yani 1 ve 0'lardan oluşan sayı gruplarına dayanmaktadır. Daha sonraları *Assembly* denen insanların daha rahat okuyabileceği ilk dil ortaya çıkmıştır. Komut ve işlenen birimden oluşan kısa ifadeler halinde olan *Assembly* dili bilgisayarların merkezi işlem birimlerinin her türü için ayrı ayrı olabilirler.

11.1.2.2. İkinci Nesil Diller

Bu diller 1950'lerden sonra geliştirilmeye başlanmış, 1960'ların sonunda ortaya çıkan modern dillerin temelini oluşturmuşlardır. İkinci nesil diller zengin kütüphaneleri, geniş kullanım alanları ile oldukça yaygın hale gelmişlerdir. Bu diller arasında FORTRAN, COBOL, ALGOL ve BASIC sayılabilir. Bunlar arasında, FORTRAN, kırk yıla yakın bir süredir özellikle matematik ağırlıklı kütüphaneleri nedeniyle problem çözme alanında ağırlıklı olarak kullanılmaktadır. FORTRAN-66 ilk standart sürüm iken FORTRAN-77 standart sürüm (ANSI) olmuştur.

COBOL, ticari alanlarda veri işleme amaçlı olarak kullanılan belirli standartlara sahip bir dil haline gelmiştir.

ALGOL, üçüncü nesil dillere öncülük eden zengin veri tipleri ve yordamsal yapılara sahip bir dildir. Avrupa'da özellikle akademik çevrelerde yaygın bir şekilde kullanılmış ancak Amerika Birleşik Devletleri'nde pek kabul görmemiştir. ALGOL-60 ve ALGOL-68 sürümleri ile daha güçlü hale getirilmiştir.

BASIC, özellikle kişisel bilgisayarlarda kolay programlama olanağı sağlamıştır. Ancak, çok çeşitli sürümleri nedeniyle fazla standart hale gelmemiştir. Günümüzde ise bu dilin grafiksel gösterim amaçlı yeni bir sürümü VISUAL BASIC adıyla halen kullanılmaktadır.

11.1.2.3. Üçüncü Nesil Diller

Bu nesil diller genellikle modern ve yapısal diller olarak adlandırılır ve üç temel bölüme ayrılırlar.

- **Genel amaçlı yüksek düzey diller**

Genel amaçlı ilk dil olarak ALGOL görülmektedir. Bundan sonra onu model olarak alan PL/1, PASCAL, MODULA-2, C ve ADA izlemiştir. Bu diller, mühendislik, bilim, gömülü sistemler ve kontrol sistemleri geliştirilmesinde yaygınlıkla kullanılmaktadır.

PL/1, gelişmiş veri yapıları, çokgörevlilik desteği, karmaşık giriş/çıkış işlemleri ve listeleme işlevleri için geniş uygulama alanına sahiptir. Bu dilden daha sonraları, PL/C isimli eğitim amaçlı, PL/M isimli mikroişlemcilerde kullanım amaçlı ve PL/S isimli sistem programlama amaçlı alt diller türetilmiştir.

PASCAL daha çok modern teknikleri öğretmek amacıyla 1970'lerin başında geliştirilmiştir. Ancak, ortaya çıktıktan sonra oldukça geniş bir uygulama alanı bulmuştur.

MODULA-2 dili PASCAL'ın gelişmiş bir hali olarak ADA dilinin bir alternatifini oluşturmuştur. Bilgi gizleme, güçlü yapılar, kuvvetli tip kontrolü ve soyutlama özelliklerine rağmen günümüzde kullanım alanı neredeyse kalmamıştır.

C programlama dili aslında işletim sistemleri için geliştirilmiştir. Ünlü Unix işletim sistemi de C ile geliştirilmiştir. Günümüzde de en yaygın ve en güçlü diller arasındadır.

ADA, Amerikan Savunma Bakanlığı'nın yazılım krizine çözüm olması amacıyla gömülü gerçek-zamanlı sistemler için geliştirilmiştir. PASCAL diline benzer yapılar yanında, paketler, kalıplar, paralel görevciler (task), kesme kotarma ve hata yakalama yapıları gibi özellikleriyle günümüzde özellikle savunma alanında yaygın ve etkin bir şekilde kullanılmaktadır. ADA-83 ilk standart iken ADA-95 ile nesneye yönelim özellikleri de eklenmiş, ADA-2005 ile özellikleri genişletilmiştir.

- **Nesneye yönelik yüksek düzey diller**

Nesneye yönelik çözümlenme ve tasarımı desteklemek üzere geliştirilmiş dillerin sayısı çok artmıştır. Ancak içlerinden yalnızca C++, Objective-C,

SMALLTALK, EIFFEL, PASCAL (nesneye yönelik sürümü), JAVA dilleri uygulama alanı bulmuşlardır. Bunlardan da C++ ve JAVA günümüzün en yaygın dilleri arasındadır. Bu dillerin ortak özellikleri sınıf tanımlamaları, kapsama ve kalıtım ile çeşitli nesneye yönelim özelliklerini desteklemeleridir. Günümüzde en yaygın olarak kullanılan dillerin başında C++ ve JAVA gelmektedir. C++'ın üstünlüğü ve yaygınlığı her zaman geçerli iken JAVA dilinin yürütme amındaki başarım yetersizlikleri halen birçok uygulama alanında kabul edilemez durumdadır.

- **Özel amaçlı diller**

Özel amaçlı diller, belirli bir uygulama tipi için tasarlanmış, alışılmışın dışında yapıları ve sözdizimlerine sahiptirler. Bu tür diller dar bir uygulama alanına sahiptirler. Ancak, belirli bir uygulama tipinin gereksinimlerini karşılamak üzere geliştirildiklerinden, isterleri tasarıma ve tasarımı koda dönüştürme işlemi daha başarılı olmaktadır. En önemli diller arasında LISP, PROLOG ve APL sayılabilir.

LISP, simge ve listelerle ardışık bilgi işleme sağlamaktadır. Özellikle yapay zeka uygulamalarında, ağaç yapıları yardımıyla problem çözme amaçlı olarak kullanılır. Özyineleme (recursion) tekniği ile uzman sistemlerin ve bu sistemler için derleyicilerin geliştirilmesinde kullanılır. LISP, bilgi tabanlı (knowledge-based) sistemlerin geliştirilmesi için gerekli olan kuralları, gerçekleri ve ilişkileri tanımlama işlerini oldukça kolay hale getirmektedir.

PROLOG uzman sistemlerin geliştirilmesinde önemli yeri olan bir dildir. Dil içinde bilgi tabanlı sistemler için gerekli olan kural, gerçek ve çıkarım tanımlamaları yanında standart veri yapıları da bulunmaktadır.

APL, diziler ve vektörlerle işlem yapmak için iyi bir gerçekleştirim aracıdır. Matematiksel problemlerin çözümlerinde bir miktar kullanım yeri bulmuştur.

11.1.2.4. Dördüncü Nesil Diller

Yazılım geliştirme tarihi boyunca programlama dillerinde hep olabildiğince çok bilgi ve işlemi kapsayarak birbirinden soyutlamak amaçlanmıştır. Birinci nesil diller makine komutları düzeyinde iken, ikinci ve üçüncü nesil dillerle gerçek anlamda bilgisayar programları oluşturmak olası hale gelmiştir. Dördüncü nesil dillerle de bu soyutlama, özellikle veri soyutlaması daha da yükselmiştir.

Dördüncü nesil diller, yordamsal olan ve olmayan yapıları birleştirici özelliklere sahiptirler. Daha açık bir deyişle, kodlayıcı, koşulları ve onlara karşılık düşen yordamsal eylemleri tanımlamakta, aynı zamanda da bu işlemin sonucunu bir başka yordamsal ayrıntı içinde kullanabilmektedir. Bu nesil dilleri birkaç grupta incelemek mümkündür:

- **Sorgulama dilleri**

Veritabanı yönetimi çok geniş bir uygulama alanında kullanılmaktadır. Özellikle veri tabanlarına erişmek, sorgulama ve raporlama yapmak üzere çeşitli

diller geliştirilmiştir. DBASE III/IV, CLIPPER, DELPHI PARADOX bu diller arasında sayılabilir. Bazı sorgulama dilleri de kullanıcının var olan veri tabanları üzerinde işlem yapabilmesini sağlarlar. SQL (Structured Query Language) en yaygın olan dillerden biridir. Bazı diller de İngilizce konuşma diline çok yaklaşımlardır.

- **Kod üreticiler**

Çoğu zaman program kodunu oluşturmak üzere üçüncü nesil bir dili kullanarak uzun satırlar yazmak gerekebilir. Kod üreticiler, kullanıcının daha üst düzey yapı ve anahtar sözcüklerle daha az miktarda kod yazarak sözdizimi tam doğru olan üçüncü nesil bir dilde kod üretilmesini sağlarlar. Günümüzde yaygın bir şekilde kullanılan bilgisayar destekli yazılım mühendisliği (Computer Aided Software Engineering - CASE) araçları (UML, STP, SDL, MDA araçları gibi), grafik tabanlı kullanıcı arayüzü geliştirme araçları, iş dünyası için çeşitli araçlar C, C++, ADA gibi dillerde kod üretmektedirler. Üretilen bu kod uygun bir derleyiciden geçirilerek yürütülebilir kod elde edilmektedir.

- **Türetme diller**

Özellikle belirli alanların sorunlarını daha kolay ve etkin bir şekilde çözebilmek için bazı dillere eklemeler ve uzantılar yapılarak yeni diller türetilmektedir. Bu dillerin kullanımı, diğer diller gibi olmakta, ancak bir *önişleyici* (preprocessor) kullanılarak yeni dilde yazılan kodun daha standart bir dildeki kod haline dönüştürülmesi, sonra da bu üretilen kodun standart bir derleyici ile derlenip yürütülebilir kod üretilmesi ilkesine dayanır. Bu şekilde, normal dillerde bulunmayan birçok özellik, dilin yapıları arasına sokularak kodlayıcılara daha geniş işlevleri olan yeni diller sunulmaktadır. Örneğin, "Real-Time C++" (RTC++), C++ diline bir uzantı olarak gerçek zamanlılığı desteklemektedir. "Mentat", yine C++ diline uzantı olarak paralel ve dağıtık programlama olanağı sunmaktadır. ADA++, ADA-83 diline nesneye yönelim ve sınıf kavramı getirmektedir. Bu tür diller, gerçek uygulamalarda ve piyasada fazla yaygın olmamakla beraber, araştırma ve akademik çalışmalarda kullanılmaktadır.

- **Diğer diller**

Karar destek, prototipleme ve belirtim gibi çeşitli amaçlara hizmet diller de bulunmaktadır. Arayüz belirtim (CORBA-IDL), toplu iş tanımlama (script, batch file) gibi işlevleri yerine getirmede kullanılan diller olduğu gibi kişisel bilgisayarlarda yaygın bir şekilde kullanılan çalışma tabloları, veritabanı sistemleri de ayrı diller kullanılmaktadırlar. SGML (Standard Generalised Markup Language) ve onun daha basitleştirilmiş bir altkütmesi olan XML (Extensible Markup Language) daha çok İnternet dünyasında kullanılan HTML (Hypertext Markup Language) için bir standart getirmek üzere tanımlanmıştır. Bu diller insanların okuyabileceği türdeki verileri biçimlendirilmiş bir şekle getirmek için kullanılır. XML yalnızca metin tabanlı verileri düzenlemek ve saklamak için etkin bir dildir. Onun dışında, makinenin okuyabileceği veriler (örneğin kayan nokta sayılar) ve matematiksel hesaplamalar için kullanılmaz.

1.1.2.5. Beşinci Nesil Diller

Geleceğe hükmedecek bu son nesil diller çeşitli araştırmalar halindedirler. Yürütül-
mekte olan bu araştırmaların sonuçları henüz yaygınlaşmamıştır. Bu tür diller yeni
oluşumları kapsayacaklardır. Örneğin, nesneye yönelik programlamadan sonra
gelişen yeni bir yaklaşım da *komuya yönelik* (subject-oriented) programlamadır. Yeni
programlama dilleri, daha fazla kapsama yaratacak, belki de bizi ardışık düşünmekten
kurtaracaktır.

8. Hafta
1. Grup
San