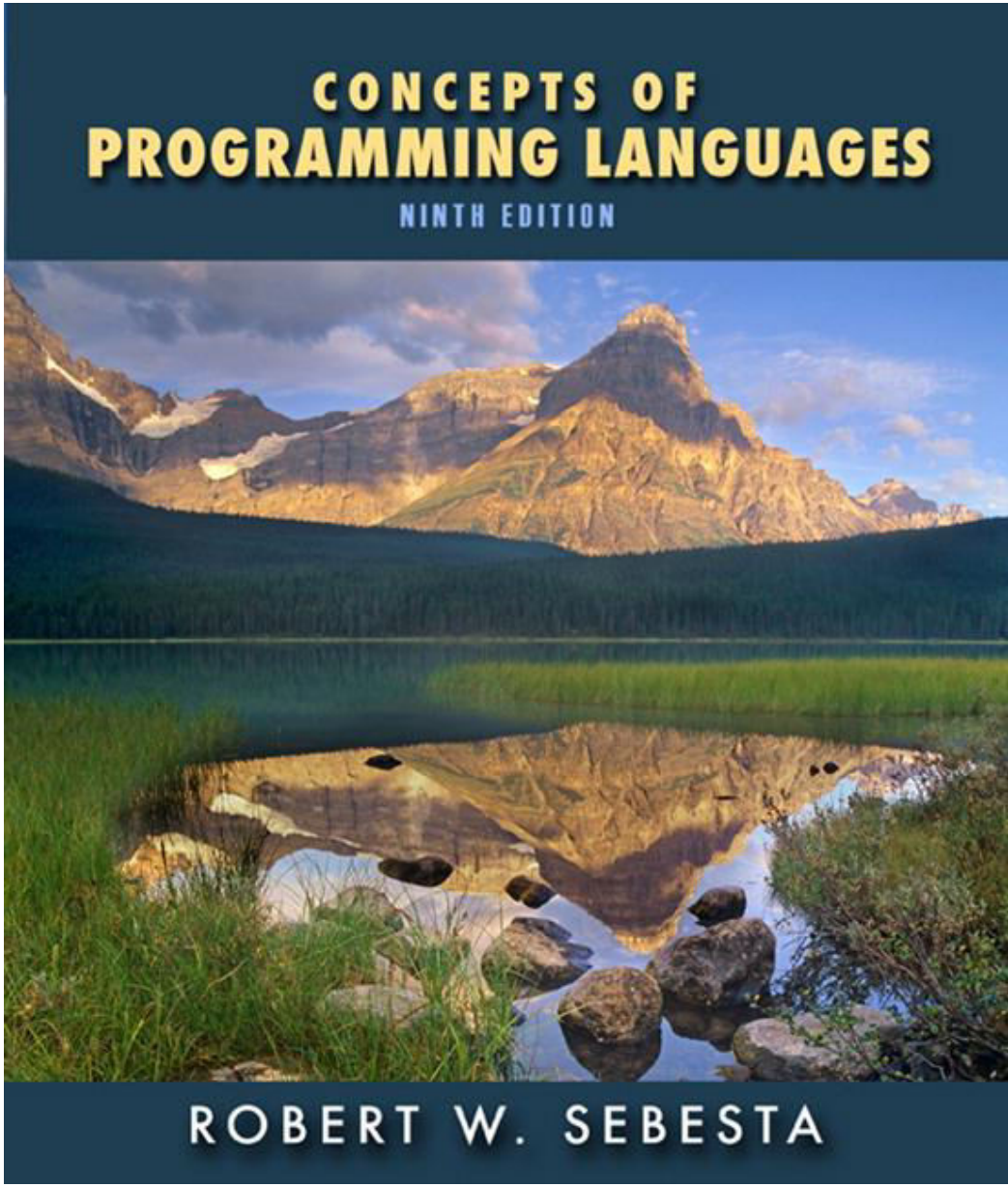
Bölüm 2

Önemli Programlama Dillerinin Gelişimi



Bölüm 2 Konular

- Zuse'nin Plankalkül'ü (Programlama dilinin adı)
- Minimum Donanımlı Programlama: Sözde kodlar (Pseudocodes)
- IBM 704 ve Fortran
- Fonksiyonel Programlama: LISP
- Sofistikeliğe (çok yönlülüğe) Doğru İlk Adım: ALGOL 60
- Ticari kayıtları Bilgisayarlaştırma: COBOL
- Zaman Paylaşımının Başlangıcı: BASIC

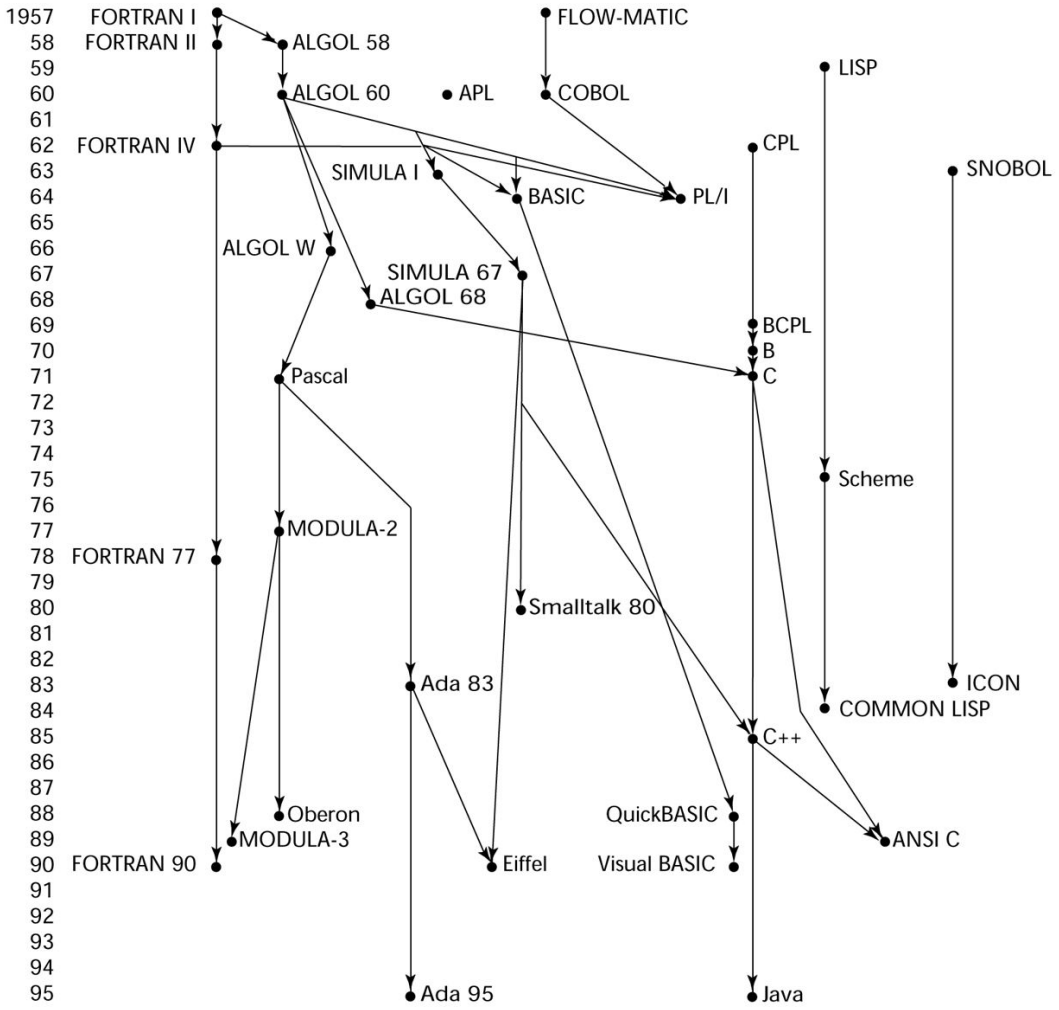
Bölüm 2 Konular (devam)

- Herkes için herşey: PL/I
- İlk İki Dinamik Dil: APL ve SNOBOL
- Veri Soyutlamanın Başlangıcı: SIMULA 67
- Ortogonal Tasarım: ALGOL 68
- Mantık Temelli Programlama: Prolog
- Tarihin en büyük tasarım çabası : Ada

Bölüm 2 Konular (devam)

- Nesne Tabanlı Programlama : Smalltalk
- Emir ve Nesne Tabanlı Özelliklerin Birleştirilmesi: C++
- Emir Temelli Nesne Tabanlı bir Dil : Java
- Metin (Script) Dilleri
- Yeni milenyum için C-temelli bir dil : C#
- Biçimlendirme / Hibrit Dillerin Programlaması

Yaygın Dillerin Şeceresi



Zuse'nin Plankalkül'ü

- 1945 yılında tasarlandı, fakat 1972 yılına kadar yayınlanmadı
- Asla uygulanmadı
- İleri veri yapıları

  –Kayan noktalı, diziler, kayıtlar
- Değişmezler

Plankalkül Sözdizimi

- A[5]'e A[4] + 1 ifadesini atamak için bir atama deyimi

```
      |  A + 1 => A
V     |  4       5        (indisler)
S     |  1.n     1.n      (veri türleri)
```

Minimum Donanım Programlama: Sözde kodlar (Pseudocodes)

- Makine kodu kullanmadaki yanlışlık neydi?
  - Zayıf okunabilirlik
  - Zayıf değiştirilebilirlik
  - İfade kodlama sıkıcı idi
  - Makine eksiklikleri – hiçbir indeksleme veya kayan nokta bulunmamaktadır

Sözde kodlar: Kısa kodlar

- Kısa kodlar 1949'da BINAC bilgisayarlar için Mauchly tarafından geliştirildi
  - İfadeler soldan sağa, kodlanmıştı
  - Örnek İşlemler:

```
01 - 06 abs value 1n (n+2)nd power
02 ) 07 +         2n (n+2)nd root
03 = 08 pause     4n if <= n
04 / 09 (         58 print and tab
```

```
X3 = ( X1 + Y1 ) / X1                    X3 03 09 X1 07 Y1 02 04 X1
```

Sözde kodlar: Hızlı kodlama

- Hızlı kodlama IBM 701 için 1954 yılında Backus tarafından geliştirilmiştir
  - Aritmetik ve matematik fonksiyonları için Sözde operatörler
  - Koşullu ve koşulsuz dallanma
  - Dizi erişimi için otomatik-artış kaydedicileri (register)
  - Yavaş!
  - Kullanıcı programı için sadece 700 kelime arta kalmaktadır

Sözde kodlar: İlgili Sistemler

- UNIVAC Sistem Derleme
  - Grace Hopper liderliğinde bir ekip tarafından geliştirildi
  - Sözdekod makine koduna genişletildi
- David J. Wheeler (Cambridge Üniversitesi)
  - Mutlak adresleme problemini çözmek için yeniden yerleştirilebilir adres blokları kullanarak bir yöntem geliştirdi.

IBM 704 ve Fortran

- Fortran 0: 1954 – uygulanmadı
- Fortran I:1957
  - Indeks kayıtlarına ve kayan nokta donanımına sahip yeni IBM 704 için tasarlanmıştır
    - Bu durum derlenmiş programlama dilleri fikrine yol açtı, çünkü yorumlama maliyetini saklayacak hiçbir yer yoktu (hiçbir kayan nokta yazılımı)
  - Gelişmenin çevresi
    - Bilgisayarlar kapasite açısından küçük ve güvenilmez idi
    - Uygulamalar bilimseldi
    - Programlama metodolojisi ve araçları yoktu
    - Makine verimliliği en önemli sorundu

Fortran Tasarım Süreci

- Fortran I tasarımında çevrenin etkisi
  - Dinamik depolamaya gerek yok
  - Iyi bir dizi işleme ve sayma döngülerine ihtiyaç duyuluyordu
  - String işleme, ondalık aritmetik, veya güçlü girdi / çıktı yoktu (iş yazılımı için)

Fortran I Genel Bakış

- Fortran'ın ilk uygulanan versiyonu
  - İsimler altı karakter uzunluğa kadar olabiliyordu
  - Post-testi sayma döngüsü (`DO`)
  - Formatlanmış I/O
  - Kullanıcı tanımlı alt programlar
  - Üç yollu seçme ifadeleri (aritmetik `IF`)
  - Veri tip ifadeleri bulunmamakta

Fortran I Genel Bakış (devam)

- FORTRAN'ın ilk uygulanan versiyonu
  - Ayrı derlemesi yoktu
  - 18 işçi-yıllık bir çaba sonrasında derleyici Nisan 1957'de çıktı
  - Genelde 704'ün kötü güvenilirliği nedeniyle 400 satırdan daha büyük programlar nadiren doğru derleniyordu.
  - Kod çok hızlı idi
  - Hızlıca yaygın olarak kullanıldı

Fortran II

- 1958'de dağıtıldı
  - Bağımsız derleme
  - Hatalar düzeltildi

Fortran IV

- 1960-62 yılları arasında gelişti
  - Açık tip bildirimleri
  - Mantıksal seçim ifadesi
  - Alt program isimleri parametreler olabiliyordu
  - 1966 yılında ANSI standardını aldı

Fortran 77

- 1978 yılında yeni standartları oldu
  - Karakter dizesi (string) işleme
  - Mantıksal döngü kontrol ifadesi
  - **`IF-THEN-ELSE`** ifadesi

Fortran 90

- Fortran 77'ye nazaran daha fazla önemli değişimler yapıldı
  - Modüller
  - Dinamik diziler
  - İşaretleyiciler
  - Önyineleme
  - **`CASE`** ifadesi
  - Parametre tür denetlemesi

Fortran'ın en son versiyonu

- Fortran 95 – nispeten ufak eklemeler, artı bazı silmeler
- Fortran 2003 – aynı

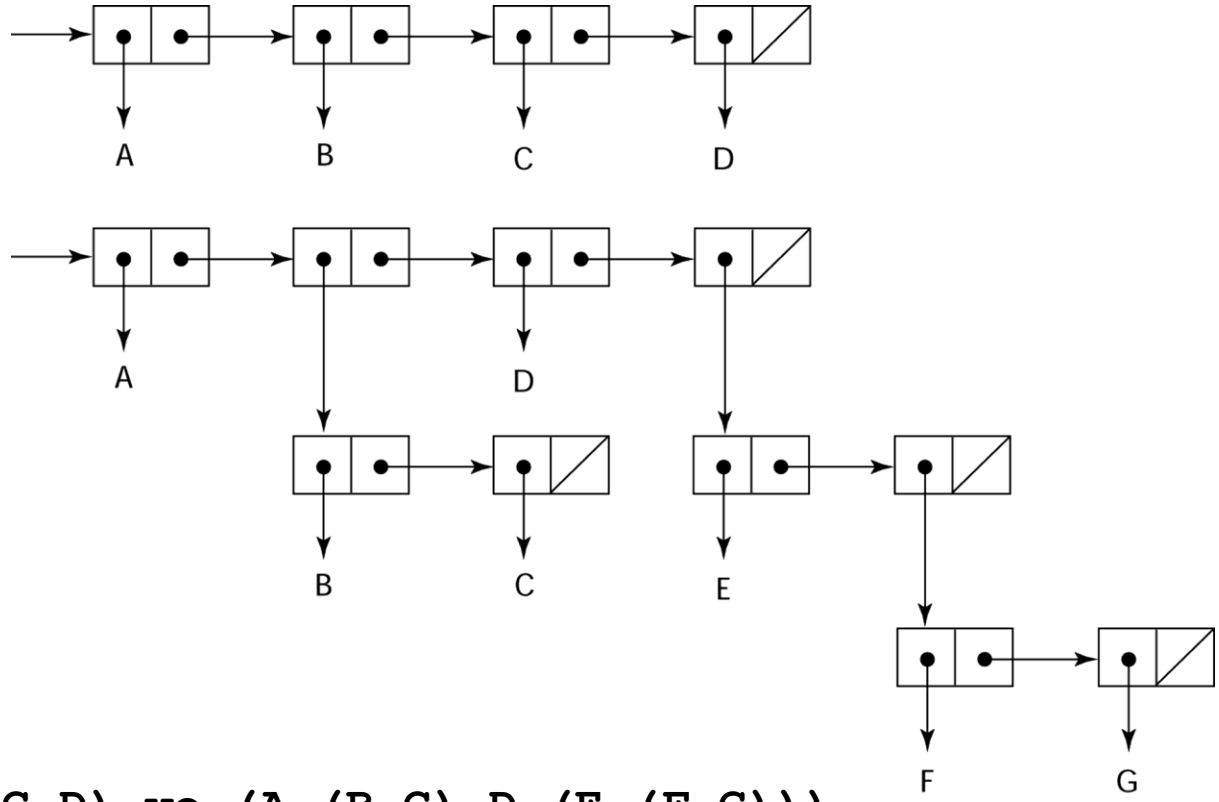
Fortran Değerlendirmesi

- Son derece optimize derleyicileri (90 öncesi tüm versiyonları)
  - Tüm değişkenlerin tip ve depolaması çalışma zamanından önce düzeltilir
- Bilgisayarların kullanıldığı yol dramatik olarak daima değişti
- Bilgisayar dünyasında geçerli dil olarak karakterize edildi

Fonksiyonel Programlama: LISP

- LISP Processing language
  - MIT'de McCarthy tarafından tasarlandı
- Yapay zeka (AI) araştırması;
  - Diziler yerine listelerde işlem verilerini,
  - Sayısal yerine sembolik hesaplamaları gerektirir
- Sadece iki veri tipi var: atomlar ve listeler
- Sözdizimi *lambda matematiğine* dayanır

İki LISP Listesinin Gösterimi



(A B C D) ve (A (B C) D (E (F G)))
Listelerinin gösterimi

LISP Değerlendirilmesi

- Öncü fonksiyonel programlama
  - Değişken ve atamaya ihtiyaç yoktur
  - Özyineleme ve koşullu ifadeler üzerinden kontrol
- AI (Artificial Intelligence) için hala baskın bir dildir
- COMMON LISP ve Scheme LISP'in çağdaş lehçeleridir

Scheme

- 1970'lerin ortalarında MIT'de geliştirildi
- Küçüktür
- Statik kapsamın yaygın kullanımı
- Birinci sınıf varlıklar olarak İşlevleri
- Basit sözdizimi (ve küçük boyutu) onu eğitim uygulamaları için ideal kılar

COMMON LISP

- LISP'in birkaç lehçesinin özelliklerini tek bir dilde birleştirme çabasıdır
- Büyük, karmaşık

Sofistikeliğe doğru İlk Adım: ALGOL 60

- Gelişme çevresi
  - FORTRAN (sadece) IBM 70x içindi
  - Diğer bir çok dil geliştirildi, tümü özel makineler içindi
  - Taşınabilir dil değillerdi; hepsi makineye bağlı idiler
  - Algoritma haberleşmesi için üniversal bir dil değildi
- ALGOL 60 üniversal bir dil tasarlama çabasının sonucu idi

Erken Tasarım Süreci

- ACM ve GAMM (27 Mayıs –1 Haziran 1958) tasarım için sadece dört gün bir araya geldi

- Dilin Hedefleri
  - Matematik notasyonlara yakın olmalı
  - Algoritmaları tanımlamak için iyi olmalı
  - Makine koduna çevrilebilmeli

ALGOL 58

- Tip kavramı resmileştirildi
- İsimler herhangi bir uzunlukta olabilirdi
- Diziler indislerin herhangi bir sayısı olabilir
- Parametreler (in & out) modu ile ayrıldı
- İndisler köşeli parantezler içine yerleştirildi
- Bileşik ifadeler (**`begin ... end`**)
- Bir deyimi ayırıcı olarak noktalı virgül kullanıldı
- Atama operatörü:= idi
- **`If`**, **`else-if`** cümlesi içerirdi
- I/O yok – "Bu durum makine bağımlı hale getirirdi"

ALGOL 58 Uygulaması

- Uygulanacağı anlamına gelmiyordu, fakat varyasyonları (MAD, JOVIAL) idi
- Başlangıçta IBM istekli olmasına rağmen, 1959 ortalarında tüm destek bırakıldı

ALGOL 60 Genel Bakış

- ALGOL 58 Paris'teki 6-gün süren toplantıda değiştirildi
- Yeni özellikleri
  - Blok yapı (yerel kapsam)
  - İki parametre geçişi yöntemleri
  - Özyineleme alt program
  - Yığın-dinamik diziler
  - Hala I / O yok ve herhangi bir dize (string) işleme bulunmamakta idi

ALGOL 60 Gelişimi

- Başarıları
  - 20 yılı aşkın sürede algoritmaları yayınlamak için standart yöntem idi
  - Müteakip emir dilleri bu (Algol 60) temelli idi
  - İlk makine bağımsız dil idi
  - Söz dizimi resmi tanımlanmış (BNF) ilk dil idi

ALGOL 60 Gelişimi (devam)

- Başarısızlıkları
  - Özellikle Amerika'da asla yaygın olarak kullanılmadı
  - Nedenleri
    - I/O eksikliği ve karakter seti eksikliği yüzünden programları taşınmaz yapmaktaydı
    - Çok fazla esnek idi–uygulama zordu
    - Fortran köklü (Entrenchment) idi
    - Örgün sözdizimi açıklama
    - IBM'den yeterli destek alamaması

Ticari Kayıtları Bilgisayarlaştırma: COBOL

- Gelişim çevresi
  - FLOW-MATIC'i kullanmak için UNIVAC başlangıç idi
  - AIMACO'yu kullanmak için USAF başlangıç idi
  - COMTRAN'ı IBM geliştirdi

COBOL'un Tarihsel Geçmişi

- FLOW-MATIC temellidir
- FLOW-MATIC özelliklerini taşır
  - İsimler 12 karaktere kadar olabiliyordu, gömülü tire ile birlikte
  - Aritmetik operatörler için İngilizce isimler (Aritmetik ifadeler yoktu)
  - Veri ve kod tümden birbirinden ayrı idi
  - Her ifade içerisindeki ilk kelime bir fiil idi

COBOL Tasarım Süreci

- İlk Tasarım Toplantısı (Pentagon) – May 1959
- Tasarım Hedefleri
  - Basit İngilizce gibi görünmeli
  - O az güçlü olacağı anlamına gelse bile, kullanımı kolay olmalı
  - Bilgisayar kullanıcıları tabanını genişletmeli
  - Mevcut derleyici problemleriyle kısıtlanmış olmamalı
- Tasarım komitesi üyelerinin tamamı bilgisayar üreticilerinden ve DoD (Dept. Of Defence) birimlerinden oluşuyordu
- Tasarım Problemleri: aritmetik ifadeler? indisler? Üreticileri arasında Savaşlar

COBOL Değerlendirmesi

- Katkıları
  - Yüksek düzeyli bir dilde ilk kez makro olanağı
  - Hiyerarşik veri yapıları (kayıtlar)
  - İç içe seçim ifadeleri
  - Tire ile uzun isimler (30 karaktere kadar),
  - Ayrı veri bölümü

COBOL: DoD Etkisi

- DoD tarafından ihtiyaç duyulan ilk dil
  - DoD desteği olmasaydı başarısız olacaktı
- Halen en yaygın kullanılan ticari uygulama dilidir

Zaman Paylaşım Başlangıcı: BASIC

- Dartmouth'da Kemeny ve Kurtz tarafından tasarlanmıştır
- Tasarım Hedefleri:
  - Öğrenmesi kolay ve fen bilgisi öğrencisi olmayanlar kullanabilmeli
  - "Hoş ve arkadaşça" olmalı
  - Ödevler hızlı yapılabilmeli
  - Ücretsiz olmalı ve kişisel erişim özelliği olmalı
  - Kullanıcı zamanının bilgisayar zamanından çok daha önemli olduğu unutulmamalı
- Mevcut popüler diyalekt: Visual BASIC
- Zaman paylaşımı ile birlikte kullanılan ilk yaygın dil

2.8 Herkes için Herşey : PL/I

- IBM ve SHARE tarafından tasarlandı
- 1964 yılındaki bilgisayar durumu (IBM'in bakış noktası)
  - Bilimsel hesaplama
    - IBM 1620 ve 7090 bilgisayarlar
    - FORTRAN
    - SHARE kullanıcı grubu
  - İş hesaplama
    - IBM 1401, 7080 bilgisayarlar
    - COBOL
    - GUIDE kullanıcı grubu

PL/I: Geçmişi

- 1963'de
  - COBOL'deki gibi Bilimsel kullanıcılar, I / O daha ayrıntılı ihtiyaç duymaya başladılar; iş kullanıcıları MIS için kayan nokta ve diziye ihtiyaç duymaya başladılar
  - Bilgisayarların iki çeşidi, dilleri ve teknik eleman desteği için çok satış yapacak gibi görünüyordu—Çok maliyetli idi
- Bariz çözüm
  - Her iki tür uygulamaları yapmak için yeni bir bilgisayar yapmak
  - Uygulamaların her iki çeşidini de yapabilecek yeni bir dil tasarlamak

PL/I: Tasarım Süreci

- 6 Komite üyesi tarafından beş ayda tasarlandı
  - IBM'den üç üye, SHARE üç üye
  - İlk kavram
  - Fortran IV'ün bir uzantısı
- Başlangıçta NPL olarak adlandırıldı (New Programming Language)
- 1965'de ismi PL/I oldu

PL/I: Değerlendirme

- PL/I katkıları
  - Birim seviyesi eşzamanlılıkta bir ilk
  - Harici işlemlerde bir ilk
  - Anahtar-seçmeli özyineleme
  - İşaretleyici veri türünde bir ilk
  - Dizi kesitlerinde bir ilk
- Endişeler
  - Birçok yeni özellik zayıf tasarlanmıştı
  - Çok büyük ve çok karmaşıktı

İlk iki Dinamik Dil : APL ve SNOBOL

- Dinamik yazma ve bellek tahsisi ile karakterize edilir
- Değişkenler yazılmaz
  - Bir değer atandığı zaman değişken tip edinir
- Bir değer atandığı zaman bellekte bir değişken tahsis edilir

APL: A Programlama Dili

- 1960 civarında Ken Iverson tarafından IBM'de çalışan bir donanım tanımlama dili olarak tasarlanmıştır
  - Çok anlamlıdır (birçok operatör, çeşitli boyutlarda hem skaler ve diziler için)
  - Programların okunması çok zor
- Halen kullanımdadır; Az düzeyde değişiklikler yapıldı

SNOBOL

- 1964 yılında Farber, Griswold ve Polensky tarafından Bell Laboratuvarları'nda metin (string) işleme dili olarak tasarlandı
- Metin desen eşleştirme için güçlü operatörlere sahip
- Alternatif dillerden (ve bu yüzden artık yazım editörleri tarafından kullanılmamaktadır) daha yavaş
- Halen bazı metin işleme görevleri için kullanılmaktadır

Veri Soyutlamanın Başlangıcı : SIMULA 67

- Nygaard ve Dahl tarafından öncelikle Norveç'te sistem simülasyonu için tasarlandı
- ALGOL 60 ve SIMULA I temellidir
- Birincil Katkıları
  - Eşyordamlar – bir çeşit alt program
  - Sınıflar, nesneler ve miras

Ortogonal Tasarım: ALGOL 68

- ALGOL 60'ın süregelen gelişmesindendir, fakat o dilin üstü değildir
- Birçok yeni fikirlerin kaynağıdır (dilin kendisi hiçbir zaman yaygın kullanıma ulaşamamasına rağmen)
- Tasarım ortogonal kavramına dayanmaktadır
  - Birkaç temel kavramlar, artı birkaç birleştirici mekanizma

ALGOL 68 Değerlendirme

- Katılımlar
  - Kullanıcı tanımlı veri yapıları
  - Referans türleri
  - Dinamik diziler (flex diziler olarak adlandırılır)
- Yorumlar
  - ALGOL 60 dan daha az kullanım
  - Müteakip dillerde güçlü etkisi oldu, özellikle Pascal, C ve Ada üzerinde

Pascal – 1971

- Wirth (o dönemlerde ALGOL 68 komitesi üyesi) tarafından geliştirildi
- Yapısal programlama öğretmek için tasarlandı
- Küçük, basit, gerçekte yeni bir şey yok
- En büyük etkisi programlama öğretme üzerine oldu
  - 1970'lerin ortalarından başlayarak 1990'ların sonlarına kadar, programlama öğretmek için kullanılan en yaygın dildi

C – 1972

- (Dennis Richie tarafından Bell Laboratuvarları'nda) sistem programlaması için tasarlandı
- Öncelikle BCLP, B, fakat aynı zamanda ALGOL 68'den geliştirildi
- Güçlü operatörler setine sahip, fakat zayıf tip kontrolü var
- Başlangıçta UNIX üzerinden yayıldı
- Birçok uygulama alanı var

Mantık Temelli Programlama : Prolog

- Kowalski (Edinburgh Üniversitesi) yardımıyla, Comerauer ve Roussel (Aix-Marseille Üniversitesi) tarafından geliştirildi
- Formel mantığa dayalıdır
- Prosedürel değildir
- Verilen sorguların doğruluğunu anlamak için bir sonuç çıkarma kullanan akıllı bir veritabanı sistemi olarak özetlenebilir
- Çok verimsiz, dar uygulama alanları var

Tarihin en büyük tasarım çabası: Ada

- Büyük tasarım çabası, yüzlerce insan ilgilendi, çok fazla paraya mal oldu ve sekiz yıllık bir süreyi aldı
  - Strawman gereksinimleri (Nisan 1975)
  - Woodman gereksinimleri (Ağustos 1975)
  - Tinman gereksinimleri (1976)
  - Ironman ekipmanları (1977)
  - Steelman gereksinimleri (1978)
- İlk programcı Augusta Ada Byron ismine izafeten Ada olarak adlandırıldı

Ada Değerlendirilmesi

- Katkıları
  - Paketler – veri soyutlaması için destek
  - Kural dışı durum işleme– özenle hazırlandı
  - Generik program birimleri
  - Aynı anda kullanım – Görev modeli ile
- Yorumlar
  - Rekabetçi tasarım
  - Yazılım mühendisliği ve dil tasarımı hakkında bilinen her şeyi kapsıyordu
  - İlk derleyiciler çok zordu; ilk gerçek kullanılabilir derleyici, dil tasarımı tamamlandıktan hemen hemen 5 yıl sonra ortaya çıkabildi

Ada 95

- Ada 95 (1988 yılında başladı)
  - Nesne tabanlı programlamada tip türetimi için destek sağladı
  - Paylaşılmış veriler için daha iyi kontrol mekanizmasına sahip idi
  - Yeni aynı anda kullanım özellikleri
  - Daha esnek kütüphane
- DoD artık ihtiyaç duymaması ve C++ nın popüler olması nedenleriyle popülerliği azaldı

Nesne Tabanlı Programlama: Smalltalk

- Xerox PARC'ta geliştirildi, başlangıçta Alan Kay ve sonradan Adele Goldberg tarafından geliştirildi
- Nesne tabanlı dilde (veri soyutlama, miras ve dinamik bağlama) ilk tam uygulama
- Grafik kullanıcı arayüzü tasarımına öncülük etti
- Nesne Tabanlı Programlamaya tanıtıldı

Emir ve Nesne Tabanlı Programlamanın Birleştirilmesi: C++

- 1980 yılında Bell Laboratuvarında Stroustrup tarafından geliştirldi
- C ve SIMULA 67'den geliştirildi
- Nesne Tabanlı Programlama özellikleri kısmen SIMULA 67'den alındı
- Özel durum işleme sağlıyor
- Hem prosedürel hem de Nesne Tabanlı Programlamayı desteklediği için büyük ve karmaşık bir dildir
- OOP ile birlikte hızla, popülerliğini arttırdı
- ANSI standardı Kasım 1997'de onaylandı
- Microsoft versiyonu (2002'de .NET piyasaya sürüldü): Yönetildi C++
  - delegeler, arayüzleri, çoklu kalıtım yok

İlgili Nesne Tabanlı Diller

- Eiffel (Bertrand Meyer tarafından tasarlandı – 1992)
  - Doğrudan başka dilden değil
  - C++'dan daha küçük ve basit, fakat hala güçlü
  - C++'ın popülerliği eksikti, çünkü birçok C++ hayranları aynı zamanda C programcılarıydı
- Delphi (Borland)
  - Nesne tabanlı programlama desteklemek için Pascal'ın ilave özelliklerini kullandı
  - C++'tan daha zarif ve güvenli

Emir Temelli Nesne Tabanlı Dil: Java

- 1990'lı yılların başında Sun tarafından geliştirildi
  - C ve C++ gömülü elektronik aletler için yeterli değildi
- C++ Temellidir
  - Önemli seviyede basitleştirilmiş (**`struct`**, **`union`**, **`enum`**, işaretçi aritmetiği ve C++'ın atama zorlamalarının yarısını kapsamıyor)
  - Sadece OOP'yi destekliyor
  - Referansları var ama işaretleyicileri yok
  - Uygulamalar için destek ve eşzamanlılık formu içerir

Java Değerlendirilmesi

- C++'ın birçok güvensiz özelliklerini bertaraf etti
- Eşzamanlılığı destekliyor
- Apletler için kütüphaneleri var, GUI, veritabanı erişimi mümkün
- Taşınabilir: Java Sanal Makine konsepti, JIT derleyicileri
- Web programlaması için çok kullanıldı
- Önceki dillere nazaran kullanımı daha hızlı arttı
- En son yeni versiyonu olan 5.0 2004 yılında piyasaya sürüldü

Web için Metin Dilleri

- Perl
  - Larry Wall tarafından tasarlandı—ilk olarak 1987'de piyasaya sürüldü
  - Değişkenler statik olarak yazılırdı, ancak dolaylı olarak deklere edilirdi
  - Üç ayırt edici ad, değişken adının ilk karakteri ile gösterilir
  - Güçlü, fakat tehlikeli
  - Web CGI programlama için yaygın kullanımı kazanmış
  - Ayrıca UNIX sistem yönetimi dil için yedekolarak kullanıldı
- JavaScript
  - Netscape ile başladı, fakat sonra Netscape ve Sun Mikro sistemleri ortak girişimi olarak devam etti
  - Genellikle dinamik HTML dökümanları oluşturmak için kullanılan bir istemci tarafı HTML içine gömülü bir betik dili şeklinde kullanıldı
  - Tamamen yorumlayıcıdır
  - Java ile olan ilişkisi sadece aynı söz dizimini kullanmasıdır
- PHP
  - PHP: Hiper metin önişlemci, Rasmus Lerdorf tarafından tasarlandı
  - Genellikle Web üzerinden form işleme ve veritabanı erişimi için kullanılan bir sunucu tarafı HTML içine gömülü bir betik dilidir
  - Tamamen yorumlayıcıdır

Web için Metin Dilleri

- Python
  - Nesne tabanlı yorumlayıcıya sahip bir metin dilidir
  - Tip kontrol edilir ama dinamik yazılır
  - CGI programlama ve form işleme için kullanılır
  - Dinamik yazılabilir, ancak tipi kontrol edilir
  - Listeleri, değişkenler gurubu ve karmaları destekler
- Lua
  - Nesne tabanlı yorumlayıcıya sahip bir metin dilidir
  - Tip kontrol edilir ama dinamik yazılır
  - CGI programlama ve form işleme için kullanılır
  - Dinamik yazılabilir, ancak tipi kontrol edilir
  - Listeleri, değişkenler gurubu ve karmaları destekler, bütün bunları onun tek veri yapısı ve tabloları üzerinden yapar
  - Kolayca genişletilebilir

Web İçin Metin Dilleri

- Ruby
  - Yukihiro Matsumoto (a.k.a, “Matz”) tarafından Japonya’da tasarlandı
  - Perl ve Python için yedek bir dil olarak başladı
  - Bir saf nesne yönelimli bir (Script) dil
    - Tüm veriler nesnedir
  - Birçok operatör kullanıcı kodu tarafından yeniden tanımlanabilen metotlar olarak uygulanır
  - Sade yorumlayıcıdır

Yeni Milenyum için C Temelli bir Dil: C#

- .NET geliştirme platformunun (2000) bir parçasıdır
- C++ , Java ve Delphi temellidir
- Bileşen tabanlı yazılım geliştirme için bir dil sağlar
- Bütün .NET dilleri genel sınıf kütüphanesi sağlayan Common Type System –Ortak Tip Sistemini (CTS) kullanır
- Yaygın kullanıma ulaşacak olması muhtemeldir

İşaretleme(Markup)/Programlama Hibrit Diller

- XSLT
  - eXtensible Markup Language (XML)(genişletilebilir işaretleme dili): bir metamarkup dili
  - eXtensible Stylesheet Language Transformation (XSTL)(genişletilebilir stilsayfası dil dönüşümü) XML dökümanlarını görüntülenebilmesi için dönüştürür
  - Programlama yapıları (örn., döngüler)
- JSP
  - Java Server Pages(Java Sunucu Sayfaları): dinamik web dökümanlarını destekleyen teknolojiler koleksiyonu
  - servlet: bir Web servera ait bir Java programı; servlet'in çıktısı browserda görüntülenir

Özet

- Geliştirme (development), geliştirme platformu (development environment), ve bazı önemli programlama dillerinin değerlendirilmesi
- Dil tasarımındaki mevcut sorunlara bakış açısı