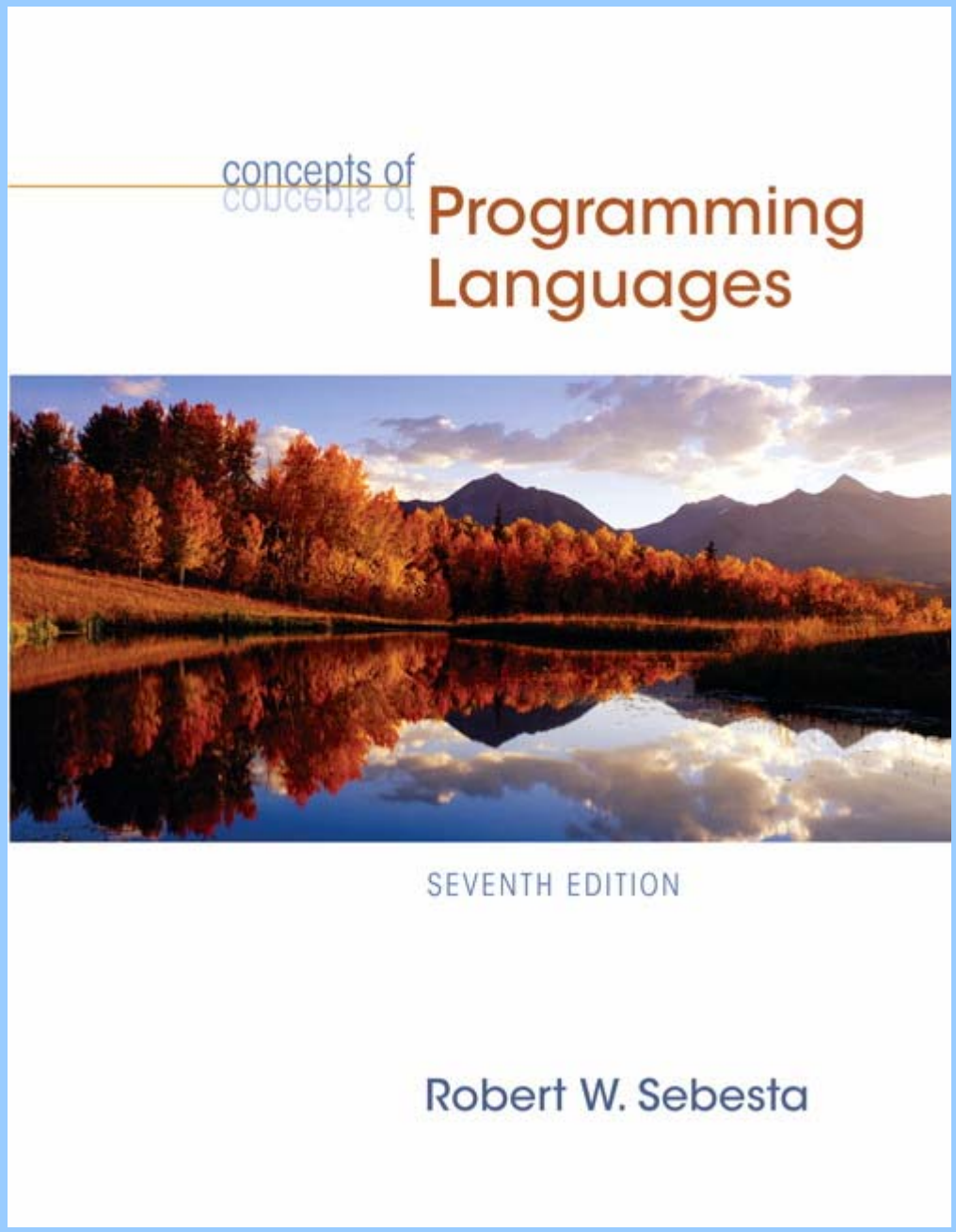
Bölüm 2

Ana Programlama Dillerinin Evrimi



ISBN 0-321-33025-0

Bölüm 2 Konular

1. Zuse'nin Plankalkul' ü
2. Minimum Donanım Programlama: Sözde kod(Pseudocode)
3. IBM 704 ve Fortran
4. Fonksiyonel Programlama: LISP
5. Sofistikeliğe doğru ilk adım: ALGOL 60
6. Ticari Kayıtları bilgisayara uyarlamak: COBOL
7. Zaman Paylaşımının(Timesharing) başlangıcı: BASIC

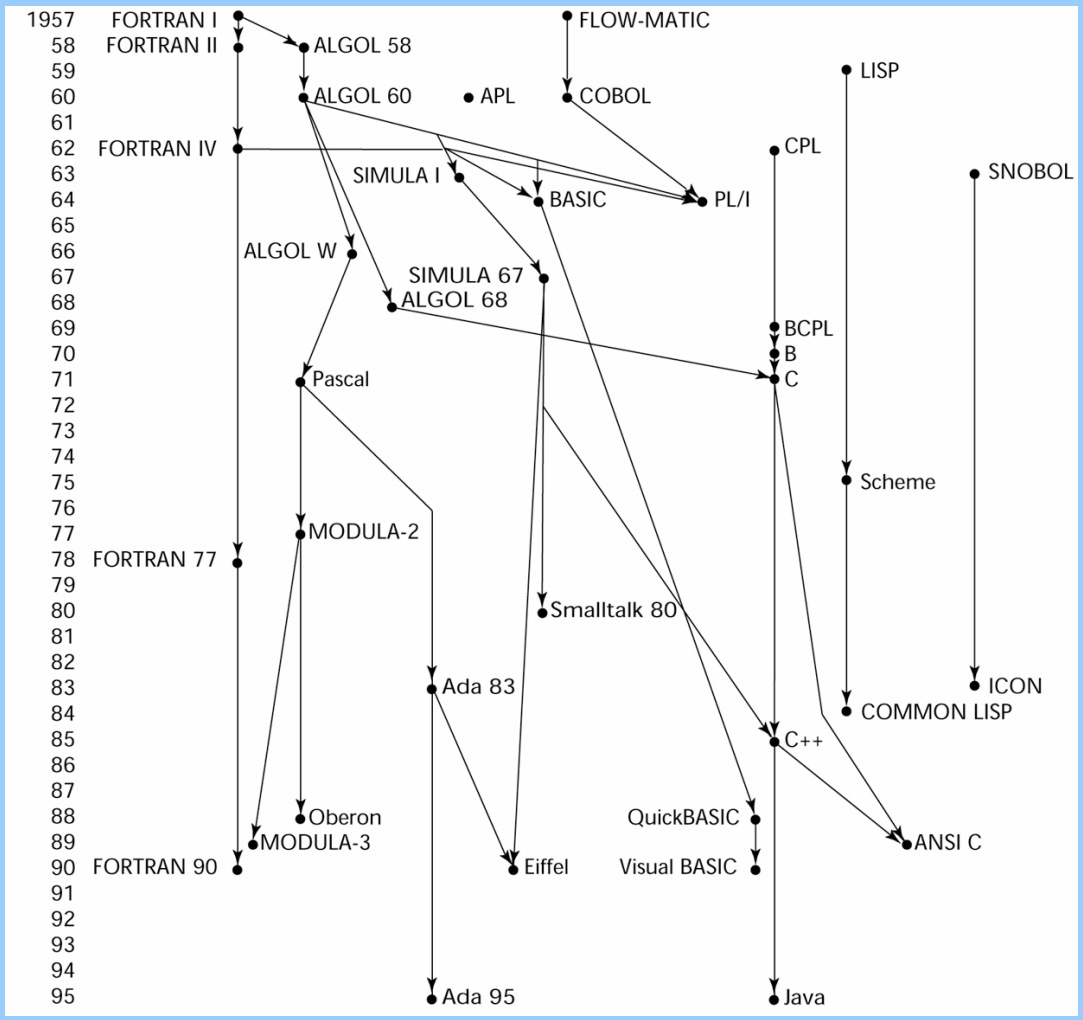
Bölüm 2 Konular (devamı)

8. Herkes için Herşey: PL/I
9. İlk iki Dinamik Dil: APL ve SNOBOL
10. Veri Soyutlama(Data Abstraction) nın başlangıçları: SIMULA 67
11. Ortogonal(Orthogonal) Dizayn: ALGOL 68
12. ALGOL'lerin ilk torunlarından bazıları
13. Mantık(Logic) temelli programlama: Prolog
14. Tarihin en büyük tasarım çabası: Ada

Bölüm 2 Konular (devamı)

15. Nesneye-dayalı Programlama: Smalltalk
16. Zorunlu(Imperative) ve nesneye-dayalı(Object-Oriented) özellikleri birleştirmek: C++
17. Bir Zorunlu nesneye-dayalı dil(Imperative-Based Object-Oriented): Java
18. Betik Diller(Scripting Languages): JavaScript, PHP, ve Python
19. Yeni milenyum için C-temelli bir dil: C#
20. İşaretleme(Markup)/Programlama Hibrit Diller

Yaygın Dillerin Soyağacı(Genealogy)



2.1 Zuse'nin Plankalkül'ü

- Asla geliştirilmedi
- İleri veri yapıları
  - Kayan nokta(floating point), diziler(arrays), kayıtlar(records)
- Sabitler(Invariants)

Plankalkül Sentaksı

- A[4] + 1 deyimini A[5] ‘e atayan bir ifade

	|	A + 1 => A	
V	|	4 5	(altsimgeler–subscripts)
S	|	1.n 1.n	(veri tipleri–data types)

2.2 Minimum Donanım Programlama: Sözdekodlar(Pseudocodes)

- Makine kodu kullanmak neden yanlıştı?
  - Az okunabilirlik
  - Az değiştirilebilirlik
  - Deyim kodlama(Expression coding) usandırıcıydı
  - Makine eksiklikleri—indeksleme veya kayan nokta(floating point) yoktu

Sözdekodlar: Short Code(Kısa Kod)

- Short Code,1949 yılında Mauchly tarafından BINAC bilgisayarları için geliştirildi
  - Deyimler(Expressions) -soldan sağa doğru- kodlandı
  - İşlemlerden örnekler:

```
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
```

Sözdekodlar: Speedcoding (hızlıkodlama)

- Speedcoding 1954 yılında Backus tarafından IBM 701 için geliştirildi
- Aritmetik ve matematiksel fonksiyonlar için sözde işlemler
  - Koşullu(conditional) ve koşulsuz(unconditional) dallanma(branching)
  - Dizi erişimi için kaydedicileri(registers) otomatik arttırır
  - Yavaştır!
  - Kullanıcı programı için sadece 700 kelime ayrılmıştır

Sözdekodlar : İlgili Sistemler

- UNIVAC Derleme Sistemi
  - Grace Hopper yönetimindeki bir ekip tarafından geliştirilmiştir
  - Sözdekod makine koduna genişletilmiştir
- David J. Wheeler (Cambridge University)
  - Salt Adresleme(absolute addressing) problemini çözmek için yeniden-yerleştirilebilir adres blokları kullanan bir metot geliştirmiştir

2.3 IBM 704 ve Fortran

- Fortran 0: 1954 - uygulanmamıştır
- Fortran I:1957
  - İndeks yazmaçları(registers) ve kayan nokta(floating point) donanımına sahip yeni IBM 704 için tasarlanmıştır
  - Geliştirme platformu
    - Bilgisayarlar küçük ve güvenilmezdi
    - Uygulamalar bilimseldi
    - Programlama metodolojileri ve araçları yoktu
    - Makine verimliliği en önemlisiydi

Fortran'ın tasarım işlemi

- Fortran I'in tasarımına platformun(environment) etkisi
  - Dinamik belleğe(storage) ihtiyaç yoktu
  - İyi dizi(array) işleme ve sayma döngülerine (counting loops) ihtiyaç vardı
  - string işleme, ondalık aritmetik, veya güçlü girdi/çıktı(ticari ürünler) yoktu

Fortran I 'e bakış

- FORTRAN'ın gerçekleştirilmiş ilk sürümü
  - İsimler altı karaktere kadar olabiliyordu
  - Test-sonrası sayma döngüsü (Post-test counting loop) (**DO**)
  - Biçimlendirilmiş Girdi/Çıktı (Formatted I/O)
  - Kullanıcı-tanımlı altprogramlar
  - Üçlü seçim ifadesi (aritmetik **IF**)
  - Veri tipi ifadeleri yoktur

Fortran I ‘e bakış (devamı)

- FORTRAN’ın gerçekleştirilmiş ilk sürümü
  - Ayrı derleme yoktur
  - Derleyici(compiler), 18 iş-yılı çabadan sonra Nisan 1957’de çıktı
  - 400 satırdan fazla programlar, 704’ün az güvenilirliği yüzünden nadiren doğru derleniyordu
  - Kod çok hızlıydı
  - Kısa zamanda yaygın kullanılır hale geldi

Fortran II

- 1958’ de yayıldı
  - Bağımsız derleme
  - Hataları(bugs) düzeltti

Fortran IV

- 1960-62 yıllarında geliştirildi
  - Belirtilmiş(explicit) tip tanımlamaları
  - Mantıksal seçim ifadesi
  - Altprogram(Subprogram) isimleri parametre olabilir
  - 1966 ANSI standardı

Fortran 77

- 1978 de yeni standart haline geldi
  - Karakter dizisi(string) işleme
  - Mantıksal döngü kontrol ifadesi
  - **`IF-THEN-ELSE`** ifadesi

Fortran 90

- Fortran 77'den en önemli farkları
  - Modüller
  - Dinamik diziler(arrays)
  - İşaretçiler(Pointers)
  - Özyineleme(Recursion)
  - **CASE** ifadesi
  - Parametre tipi testi (parameter type checking)

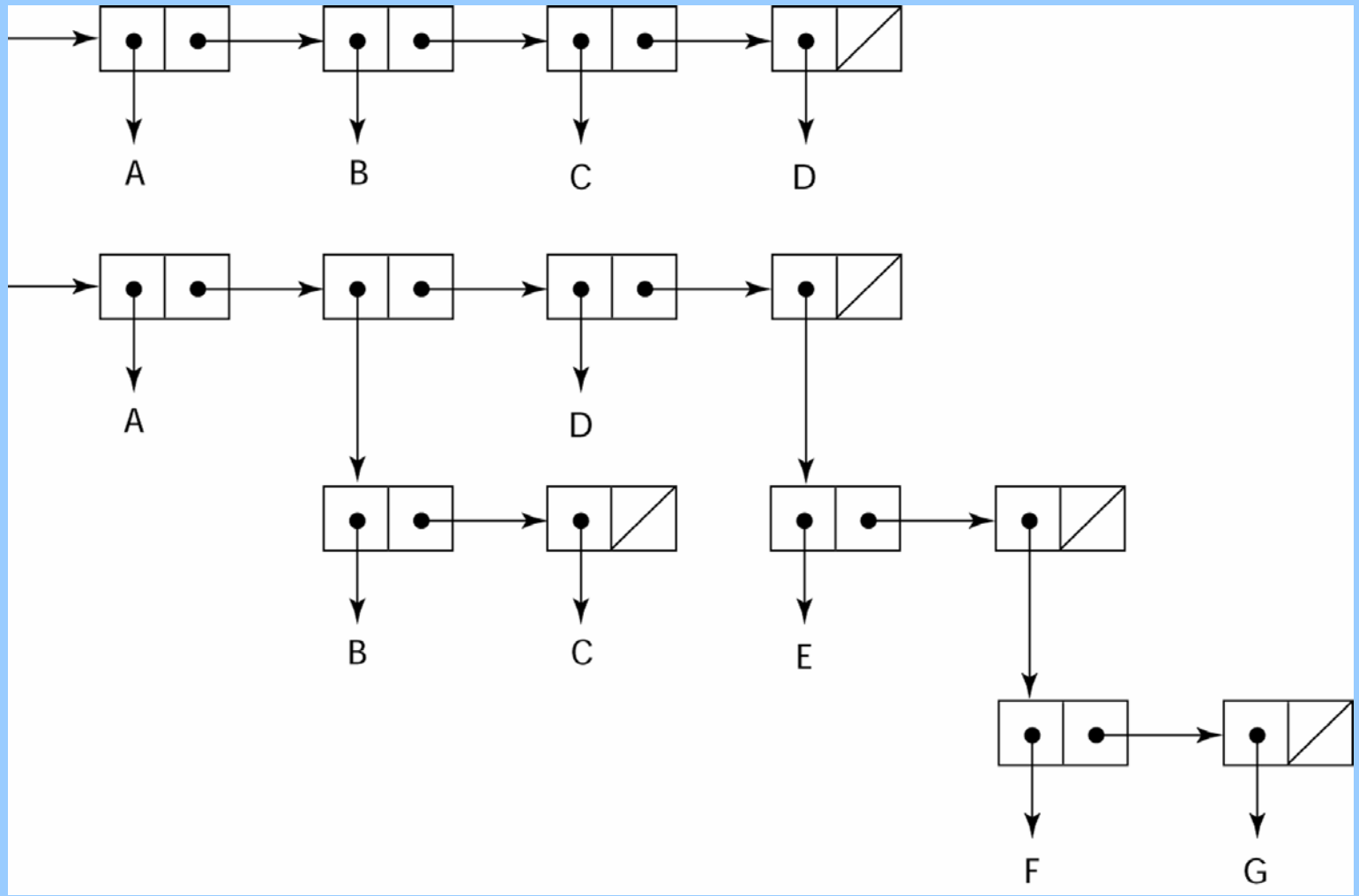
Fortran Değerlendirmesi

- Çok iyi optimize eden derleyiciler (90'dan önceki tüm sürümler)
  - Bütün değişkenlerin tipleri ve bellekleri çalışma zamanından(run-time) önce düzeltilir
- Bilgisayarların kullanılma şeklini sürekli çarpıcı biçimde değiştirdi
- Bilgisayar dünyasının *lingua franca (uluslarası dil)* 'sı olarak karakterize edildi

2.4 Fonksiyonel Programlama: LISP

- LİSte işleme dili
  - McCarthy tarafından MIT'de tasarlandı
- AI (Artificial Intelligence-yapay zeka) araştırmasının ihtiyaç duyduğu dil şöyleydi:
  - Veriyi liste halinde işleme (dizi(array) yerine)
  - Sembolik hesaplama (sayısal yerine)
- Sadece iki veri tipi: *atom*lar and *list* (e)ler
- Sentaks *lambda calculus*'a dayalıdır

İki LISP Listesini Gösterimi



LISP Değerlendirmesi

- Fonksiyonel programlamada öncü olmuştur
  - Değişkenlere(variables) veya atamaya(assignment) ihtiyaç yoktur
  - Kontrol, özyineleme(recursion) ve koşullu ifadeler(conditional expressions) ile sağlanır
- Halen AI için hakim olan dildir
- COMMON LISP ve Scheme, LISP'in güncel diyalektleridir(lehçeleridir-versiyonlarıdır)
- ML, Miranda, ve Haskell ilgili dillerdir

Scheme

- MIT ‘de 1970’lerin ortalarında geliştirildi
- Küçüktür
- Statik kapsam(static scoping)’ın geniş kullanımı
- Fonksiyonlar birinci-sınıf varlıklardır(first-class entities)
- Basit sentaksı (ve küçük boyutu), eğitim amaçlı uygulamalar için ideal olmasını sağlamıştır

COMMON LISP

- LISP'in birkaç diyalektinin özelliklerini bir dilde toplama gayretidir
- Büyük, karmaşık

2.5 Sofistikeliğe doğru ilk adım: ALGOL 60

- Geliştirme platformu
  - FORTRAN ancak IBM 70x içindi
  - Geliştirilmekte olan diğer bütün diller belirli makineler içindi
  - Taşınabilir dil yoktu; hepsi makine-bağımlıydı
  - Haberleşme algoritmaları için evrensel bir dil yoktu
- ALGOL 60 evrensel bir dil tasarlama çabalarının sonucuydu

İlk tasarım işlemi

- ACM ve GAMM tasarım için dört gün görüştü (Mayıs 27 den Haziran 1'e, 1958)
- Dilin amaçları
  - Matematiksel gösterime yakın
  - Algoritma tanımlamak için iyi
  - Makine koduna çevrilebilir olmalıydı

ALGOL 58

- Tip kavramı resmileştirildi
- İsimler herhangi bir uzunlukta olabilirdi
- Diziler(arrays) herhangi bir sayıda altsimgeye(subscripts) sahip olabilirdi
- Parametreler kip(mode) ile ayrıldı (in & out)
- Altsimgeler(Subscripts) köşeli parantezler içine yerleştirilmişti
- Bileşik(compound) ifadeler (**`begin ... end`**)
- Noktalı virgül ayırıcı olarak kullanıldı
- Atama işleci := oldu
- **`if`** 'in **`else-if`** deyimi vardı
- I/O yoktu – "onu makine bağımlı hale getirirdi"

ALGOL 58 Implementasyon

- Geliştirmesi planlanmadı, fakat çeşitleri şunlardı:(MAD, JOVIAL)
- Başlangıçta IBM istekli olmasına rağmen, 1959 ortalarında tüm destek geri çekildi

ALGOL 60 ‘e bakış

- Paris’teki 6-günlük toplantı sonucunda ALGOL 58’in değiştirilmesiyle geliştirildi
- Yeni özellikler
  - Blok yapısı (yerel kapsam--local scope)
  - İki parametre geçişi(parameter passing) metodu
  - Altprogram özyineleme(recursion)
  - Yığın-dinamik diziler(Stack-dynamic arrays)
  - Hala I/O (girdi/çıktı) ve dizim(string) işleme yoktu

ALGOL 60 Değerlendirmesi

- Başarıları
  - 20 yılı aşkın süre algoritma yayınlamanın standart yolu olarak kalmıştır
  - Sonra gelen bütün zorunlu(imperative) diller ona dayandırılmıştır
  - İlk makine-bağımsız dildir
  - Sentaksı resmi olarak tanımlanan ilk dildir (BNF)

ALGOL 60 Değerlendirmesi (devamı)

- Başarısızlıkları
  - Hiçbir zaman, özellikle U.S.'de yaygın olarak kullanılmamıştır
  - Nedenleri
    - I/O ve karakter kümesinin eksikliği programları taşınamaz yapmıştır
    - Aşırı esnek—geliştirilmesi zor
    - Fortran'ın sağlamlaştırılması
    - Resmi sentaks tanımı
    - IBM desteğinin eksikliği

Ticari Kayıtları bilgisayara uyarlamak: COBOL

- Geliştirme platformu
  - UNIVAC, FLOW-MATIC'i kullanmaya başlıyordu
  - USAF, AIMACO' yu kullanmaya başlıyordu
  - IBM , COMTRAN'ı geliştiriyordu

COBOL Tarihi arkaplan

- FLOW-MATIC temellidir
- FLOW-MATIC özellikleri
  - İsimler gömülü tirelerle(kısa çizgi- hyphen) 12 karaktere kadar çıkabiliyordu
  - Aritmetik operatörler için İngilizce isimler (aritmetik deyimler yoktu)
  - Veri ve kod tamamen ayrıydı
  - Her ifadede fiil(verb) ilk kelimeydi

COBOL Tasarım İşlemi

- İlk tasarım toplantısı (Pentagon) – Mayıs 1959
- Tasarım amaçları
  - Basit İngilizce gibi görünmeli
  - Daha az güçlü olacağı anlamına gelse bile kullanımı kolay olmalı
  - Bilgisayar kullanıcıların tabanını genişletmeli
  - Mevcut derleyici problemleriyle kısıtlanmış olmamalı
- Tasarım komitesi üyelerinin tamamı bilgisayar üreticilerinden ve DoD(Amerikan savunma bakanlığı) birimlerinden oluşuyordu
- Tasarım Problemleri: aritmetik ifadeler? Altsimgeler(subscripts)? Üreticiler arasında kavgalar

COBOL Değerlendirmesi

- Katkılar
  - Bir yüksek-düzeyli dilde ilk kez makro olanağı
  - Hiyerarşik veri yapıları (records)
  - İçiçe(nested) seçim ifadeleri
  - Uzun isimler (30 karaktere kadar), tirelerle birlikte
  - Ayrı veri bölümü(data division)

COBOL: DoD Etkisi

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

2.7 Zaman Paylaşımının(Timesharing) başlangıcı: BASIC

- Dartmouth'da Kemeny ve Kurtz tarafından tasarlanmıştır
- Tasarım amaçları:
  - Bilim dışı öğrencilerin öğrenmesi ve kullanması kolay olan bir dil
  - "güzel ve arkadaşça" olmalı
  - Ödev için hızlı çalıştırılabilir olması
  - Ücretsiz ve kişisel erişim
  - Kullanıcının zamanı bilgisayarın zamanından değerlidir
- Mevcut popüler diyalekt: Visual BASIC
- Süre paylaşımlı(time sharing) ilk yaygın kullanılan dil

2.8 Herkes için Herşey: PL/I

- IBM ve SHARE tarafından tasarlandı
- 1964'te bilgisayar kullanma durumu (IBM'in bakış açısına göre)
  - Bilimsel kullanım
    - IBM 1620 ve 7090 bilgisayarları
    - FORTRAN
    - SHARE kullanıcı grubu
  - Ticari kullanım
    - IBM 1401, 7080 bilgisayarları
    - COBOL
    - GUIDE kullanıcı grubu

PL/I: Arkaplan

- 1963'le birlikte
  - Bilimsel kullanıcılar COBOL'deki gibi daha çok ayrıntılı I/O'ya; ticari kullanıcılar ise daha çok kayan nokta(floating point) ve diziye ihtiyaç duymaya başladılar
  - Öyle görünüyordu ki birçok mağaza iki çeşit bilgisayara, dile ve destek personeline ihtiyaç duymaya başlıyordu--çok pahalı
- Açıkça görülen çözüm
  - Uygulamaların iki çeşidini de yapabilecek yeni bir bilgisayar yapmak
  - Uygulamaların ikisini de yapabilecek yeni bir dil tasarlamak

PL/I: Tasarım İşlemi

- 3 X 3 Committee tarafından beş ayda tasarlandı
  - IBM'den üç üye, SHARE'den üç üye
- İlk kavram
  - Fortran IV'ün bir uzantısı
- Başlangıçta NPL (New Programming Language) adı verildi
- 1965' de adı PL/I olarak değiştirildi

PL/I: Değerlendirmesi

- PL/I katkıları
  - İlk birim-düzeyli eş zamanlı olma (unit-level concurrency)
  - İlk istisna işleme(exception handling)
  - Anahtar-seçmeli (Switch-selectable) özyineleme(recursion)
  - İlk işaretçi(pointer) veri tipi
  - İlk çapraz dizi bölümleri (array cross sections)
- Zayıflıkları
  - Birçok yeni özellik zayıf tasarlanmıştı
  - Aşırı geniş ve aşırı karmaşıktı

2.9: İlk iki Dinamik Dil: APL ve SNOBOL

- Dinamik tip belirleme(dynamic typing) ve dinamik bellek ayrımı (storage allocation) ile karakterize edilir
- Değişkenlerin tipi yoktur
  - Değişkene bir değer atandığı zaman değişken tip edinir
- Bir değişkene değer atandığı zaman bellekte ona yer ayrılır

APL: A Programming Language (Bir programlama dili)

- 1960'larda IBM de Ken Iverson tarafından bir donanım tanımlama dili olarak tasarlanmıştır
  - Çok anlamlıdır (hem skaler(sayısal) hem de çeşitli boyutlarda diziler için birçok operatör)
  - Programların okunması çok zordur
- Halen kullanımdadır; çok küçük değişiklikler vardır

SNOBOL

- Bell Laboratuvarlarında Farber, Griswold, ve Polensky tarafından string işleme dili oalarak tasarlanmıştır
- String desen-eşleştirme(pattern matching) için güçlü operatörler
- Alternatif dillerden daha yavaştır (ve bu yüzden artık yazım editörleri tarafından kullanılmamaktadır)
- Halen bazı metin işleme işleri için kullanılır

2.10 Veri Soyutlama(Data Abstraction) nın başlangıçları: SIMULA 67

- Norveç'de Nygaard ve Dahl tarafından asıl olarak sistem simülasyonu için tasarlanmıştır
- ALGOL 60 ve SIMULA I 'e dayalıdır
- Birincil Katkıları
  - Eş yordam(Co-routines) – bir çeşit alt program
  - Sınıf(class) adı verilen bir yapı içerisinde geliştirilmiştir
  - Sınıflar veri soyutlamanın(data abstraction) temelleridir
  - Sınıflar hem lokal veri hem de fonksiyonellik içeren yapılardır

2.11 Ortogonal(Orthogonal) Dizayn: ALGOL 68

- ALGOL 60'ın devam eden gelişmesinden meydana gelmiştir fakat onun üstkümesi değildir
- Bazı yeni fikirlerin kaynağıdır (dilin kendisinin hiçbir zaman yaygın kullanıma ulaşamamasına rağmen)
- Tasarım ortogonallik(orthogonality) kavramına dayanır
  - Birkaç prensip kavram, birkaç birleştirici mekanizma

ALGOL 68 Değerlendirmesi

- Katkılar
  - kullanıcı-tanımlı veri yapıları
  - Referans tipleri
  - Dinamik diziler (flex(esnek) arrays)
- Yorumlar
  - ALGOL 60 dan daha az kullanım
  - Sonra gelen dillerde çok etkisi olmuştur, özellikle Pascal, C, ve Ada

2.12 ALGOL'lerin ilk torunlarından bazıları

- ALGOL dilleri bütün zorunlu(imperative) dilleri etkiledi
  - Pascal
  - C
  - Modula/Modula 2
  - Ada
  - Oberon
  - C++/Java
  - Perl (bir yere kadar)

Pascal – 1971

- Wirth tarafından geliştirildi (ALGOL 68 komitesi üyesi)
- Yapısal programlama (structured programming) öğretmek için tasarlandı
- Küçük, basit, yenilik getirmeyen bir dil
- Programlama öğretmede en çok etkisi oldu
  - 1970lerin ortalarından 1990 ların sonlarına kadar, programlama öğretmek için kullanılan en yaygın dildi

C – 1972

- Sistem programlama için tasarlandı (Bell Laboratuvarlarında Dennis Richie tarafından)
- Temel olarak BCLP, B'den, aynı zamanda ALGOL 68'den geliştirildi
- Güçlü operatörler , fakat zayıf tip kontrolü (type checking)
- Başlangıçta UNIX üzerinden yayıldı
- Birçok uygulama alanı

Perl

- ALGOL ile sadece C üzerinden ilişkilidir
- Bir yazı(scripting) dilidir
  - Bir yazı dosyası(*script* file) çalıştırılacak komutları içerir
  - Diğer örnekler: sh, awk, tcl/tk
- Larry Wall tarafından geliştirilmiştir
- Perl değişkenleri statik tiplidir ve örtülü(implicitly) tanımlanmıştır
  - Üç farklı isim alanı(namespace), bir değişkenin adının ilk karakteriyle gösterilir
- Güçlü fakat tehlikeli
- Genel amaçlı bir dil olarak yaygın kullanılmaktadır

2.13 Mantık(Logic) temelli programlama: Prolog

- Comerauer ve Roussel (University of Aix-Marseille), Kowalski (University of Edinburgh) nin yardımıyla geliştirilmiştir
- Biçimsel mantığa(formal logic) dayalıdır
- Prosedürel değildir
- Verilen sorguların(query) doğruluğunu anlamak için bir sonuç çıkarma kullanan akıllı bir veritabanı sistemi olarak özetlenebilir
- Çok verimsiz, dar uygulama alanları

2.14 Tarihin en büyük tasarım çabası: Ada

- Yüzlerce insan, çok para, ve yaklaşık sekiz yıl içeren muazzam tasarım çabası
  - Strawman gereksinimleri (Nisan 1975)
  - Woodman gereksinimleri (Ağustos 1975)
  - Tinman gereksinimleri (1976)
  - Ironman gereksinimleri (1977)
  - Steelman gereksinimleri (1978)
- İlk programcı olarak bilinen Augusta Ada Byron'dan sonra Ada adı verildi.

Ada Değerlendirmesi

- Katkılar
  - Paketler – veri soyutlama desteği
  - İstisna İşleme(Exception handling) – ayrıntılı
  - Soysal(Generic) program birimleri
  - Eş zamanlılık(Concurrency) – görevleme(tasking) modeli ile
- Yorumlar
  - Rekabetçi tasarım
  - Yazılım mühendisliği ve dil tasarımı hakkında sonradan bilinen her şeyi içeriyordu
  - İlk derleyiciler çok zordu; ilk gerçekten kullanılabilen derleyici dil tasarımının tamamlanmasından yaklaşık beş yıl sonra geldi

Ada 95

- Ada 95 (1988 de başladı)
  - Tip türetme(type derivation) üzerinden OOP desteği
  - Paylaşılan veri için daha iyi kontrol mekanizmaları
  - Yeni eş zamanlılık(concurrency) özellikleri
  - Daha esnek kütüphaneler
- Popülerliği azaldı çünkü DoD daha fazla kullanımına ihtiyaç duymadı buna karşın C++ ‘ın popülaritesi arttı

2.15 Nesneye-dayalı Programlama: Smalltalk

- Xerox PARC'da, önce Alan Kay, sonra Adele Goldberg tarafından geliştirildi
- Bir nesneye-dayalı dilin ilk tamamen implementasyonu (veri soyutlama, miras(inheritance), ve dinamik tip bağlama(binding))
- Grafiksel kullanıcı arayüzü tasarımına öncülük etmiştir
- OOP 'yi yükseltmiştir

2.16 Zorunlu(Imperative) ve nesneyedayalı (Object-Oriented) özellikleri birleştirmek: C++

- Bell Labs'da Stroustrup tarafından in 1980 de geliştirilmiştir
- C ve SIMULA 67 den geliştirlmiştir
- Nesneya dayalı programlama olanakları, kısmen SIMULA 67'den alınmıştır
- İstisna yakalama(exception handling) sağlar
- Hem prosedürel hem de OO programlamayı desteklediği içinGeniş ve karmaşık bir dildir.
- Popülaritesi OOP ile birlikte hızla artmıştır
- ANSI standardı Kasım 1997'de onaylandı
- Microsoft'un sürümü(.NET le 2002 de çıkan): Yönetilmiş(Managed) C++
  - Delegeler(delegates), arayüzler(interfaces), çoklu miras (multiple inheritance) yoktur

İlgili OOP Dilleri

- Eiffel (Bertrand Meyer – 1992 de)
  - Direk olarak başka bir dilden geliştirilmemiştir
  - C++'tan küçük ve basittir, ama halen daha güçlüdür
  - C++ kadar popüler değildir çünkü birçok C++ hayranı önceden C programcılarıydı
- Delphi (Borland)
  - Pascal artı OOP yi destekleyen özellikler
  - C++'tan daha zarif ve güvenlidir elegant

2.17 Bir Zorunlu nesneye-dayalı dil (Imperative-Based Object-Oriented): Java

- 1990'ların başında Sun'da geliştirildi
  - C ve C++ gömülü electronik aygıtlar için yeterince memnun edici değildi
- C++ temellidir
  - Önemli derecede basitleştirilmiştir (**`struct`**, **`union`**, **`enum`**, işaretçi(pointer) aritmetiği, ve C++'ın atama zorlamalarının yarısını içermez)
  - *Sadece* OOP yi destekler
  - Referansları vardır, işaretçiler(pointers) yoktur
  - appletler ve bir eş zamanlılık(concurrency) formu için destek içerir

Java Değerlendirmesi

- C++'ın güvensiz özelliklerini elemiştir
- Eş zamanlılık(Concurrency) özellikleri
- Appletler için kitaplıklar, GUller, veritabanı erişimi
- Taşınabilir: Java Sanal(Virtual) Makinesi kavranı, JIT derleyiciler
- WWW sayfaları için yaygın kullanılmaktadır
- Diğer alanlarda kullanımı başka dillere oranla daha hızlı artmaktadır
- En güncel sürümü, 5.0, 2004'te çıkmıştır

2.18 Ağ(Web) için Betik Diller(Scripting Languages):

- JavaScript
  - Netscape ve Sun Microsystems ortaklığı
  - Web programlamada (istemci tarafı-client side) dinamik HTML dökümanları oluşturmak için kullanılır
  - Java ile sadece benzer sentaksı nedeniyle ilgilidir
- PHP
  - PHP: Hypertext Preprocessor
  - Web uygulamaları (sunucu tarafı-server side) için kullanılır; çıktı olarak HTML kodu üretir
- Python
  - OO yorumlanmış yazı(scripting) dilidir
  - Tip kontrol edilir ama dinamik olarak tip verilir CGI ve form işlemeyi destekler

2.19 Yeni milenyum için C-temelli bir dil: C#

- .NET geliştirme platformunun bir parçasıdır
- C++ , Java, ve Delphi temellidir
- Bileşen-temelli(component-based) yazılım geliştirme için dil sağlar
- Bütün .NET dilleri (C#, Visual BASIC.NET, Managed C++, J#.NET, ve Jscript.NET) Ortak Tip Sistemi(Common Type System (CTS)) kullanır, bu ortak bir sınıf kütüphanesi sağlar
- Yaygın kullanıma ulaşacak olması muhtemeldir

2.20 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