

BSM208

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ

Programlama Dillerinin
Tarihi Gelişimi
(Hafta2)

Programlama Dillerinin Tarihi

- 1940'ların sonu ve 1950'lerin başında;
 - Programlama dillerinde; Yavaşlık, yetersiz güvenilirlik, pahalılık, sınırlı bellek ve yazılım ve mutlak adresleme problemi bulunmaktadır.
 - Dizi indeksleme ve kayan noktalı aritmetik kavramları henüz desteklenemiyor.
 - Kayan noktalı aritmetik için donanım yetersizliğinden Interpretive sistemler tolere ediliyor ve kayan noktalı aritmetik için yazılımların yürütülmesi hala çok yavaş olmakta.

1946-1950

- İşletim sistemi kavramı yok
- Bilgisayarın yapısının çok iyi bilinmesi gerekiyor.
- Programlama=Çözülecek probleme ilişkin devre tasarlamak
- Dolayısıyla sadece uzmanlar programlama yapabilir.
(Elektronik Müh.)

BU DURUM BİLGİSAYARIN YAYGIN KULLANIMI KONUSUNDA
CİDDİ BİR PROBLEM OLUŞTURUYORDU

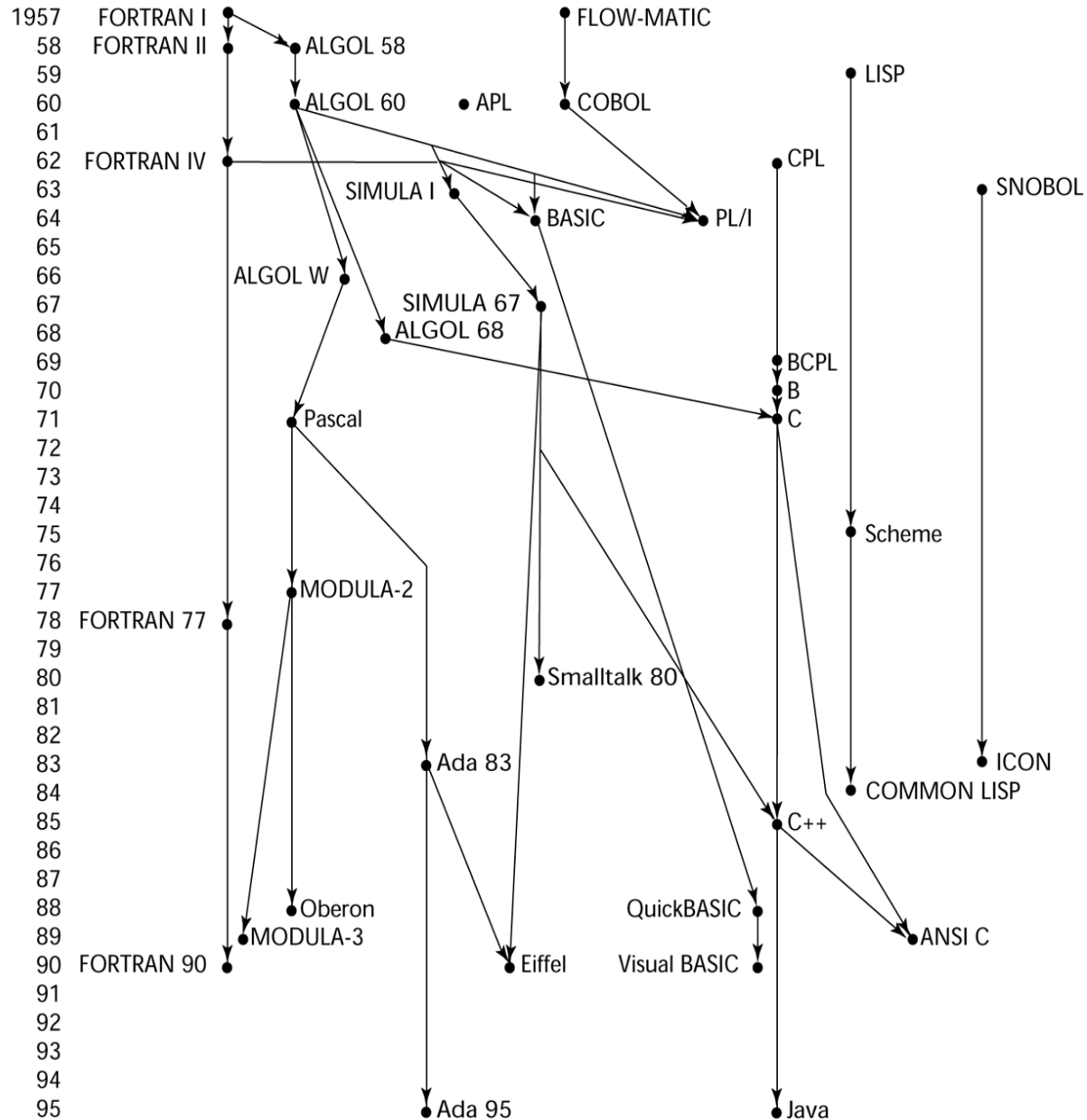
Çözüm Önerileri neydi?

- Günlük dilde kullanılan sözcüklere benzer sözcüklerden oluşan Bilgisayar dili oluşturmak
- Bu dilin anlaşılması ve öğrenilmesi kolay olmalı
- Matematiksel uygulamalar için uygun olmalı

YANI YÜKSEK SEVİYELİ BİR DİL OLMALI

Bu amaçlarla geliştirilen ilk dil FORTRAN idi

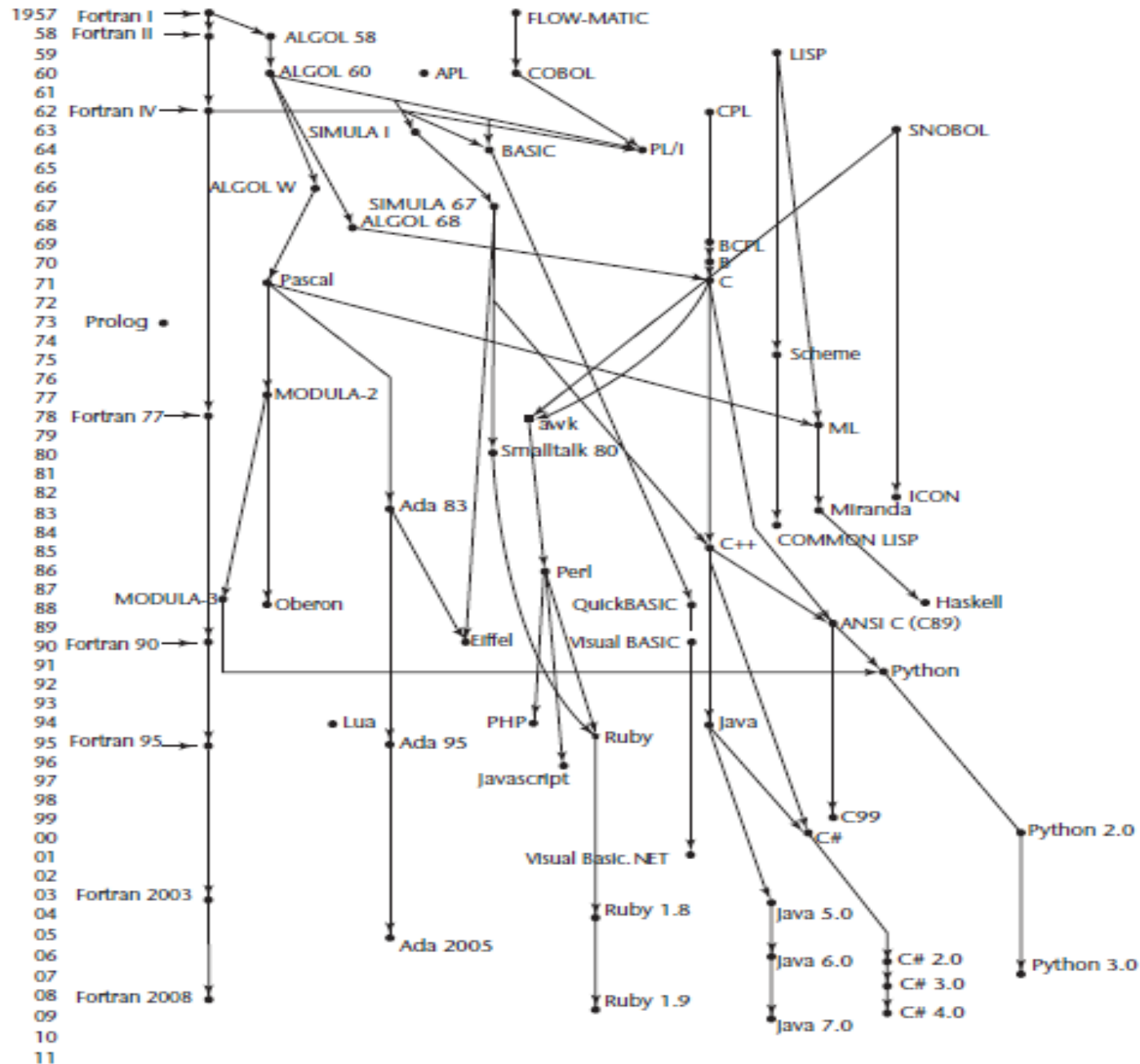
Yüksek Seviyeli Programlama Dillerinin Kronolojisi



Fortran

- FORMul TRANslating System
- 1954 yılında, IBM firmasında John Backus tarafından geliştirildi
- DO deyimleri, G/Ç deyimleri ve atama deyimleri vardı
- Matris, denklem sistemleri ve diff. Denklem sistemlerini kolaylaştırıyordu
- 3 dala sahip **If** yapısı içeriyordu:
- ***If (arithmetic_expression) N1, N2, N3***
- iterative (loop) **Do** ifadesi şu biçimde kullanılıyordu:
Do N1 variable = first_value, last_value
- Çalışma sırasında veri tipleme ifadeleri ve bellek tahsisi yoktu

Evolution of the Major Programming Languages



Fortran (devam)

- Fortran ingilizce sözcüklerden oluşuyordu
- Anlaşılması makine diline göre çok kolaydı
- Matematiksel uygulamalar için iyi özelliklere sahipti.
- Fakat;
- Fortran dilinin bilgisayar tarafından anlaşılması için DERLEYİCİ'ye ihtiyacı vardı.
- John Backus, 2 yılda, 51000 satırlık makine kodundan oluşan ve bir teyp biriminde saklanan Fortran derleyicisini üretmiştir.
- FORTRAN 1957 yılında ticari olarak kullanıma sunuldu
- Aynı kişi 1959 yılında, yüksek seviyeli bir dilin yazış kurallarını açıklama noktasında standart bir notasyon haline gelen **B**ackus **N**aur **F**orm (BNF)'yi bulmuştur.

FORTRAN ailesi

- FORTRAN I:Taşınabilirliği çok kötü
- FORTRAN 66:Karakter türü verileri işlemede çok kısıtlı, yapısal programlama yok, rekürsif işlemler yapılamıyor.
- FORTRAN 77(American National Standarts Institute-ANSI): Karakter türü verileri işleyebiliyor, Yapısal programlamayı destekliyor, DO çevrimine dışarıdan veri girilebiliyor. Taşınabilirlik daha iyi hale getiriyor.
- FORTRAN 90:Pointer (bağlı liste gerçeklemesi, dinamik bellek yönetimi), rekürsif işlem yeteneği, bit düzeyinde işlem, Dizi yapıları daha iyi kullanılabiliyor, Altprogram yapıları daha esnek, isimler daha uzun yazılabiliyor (okunabilirlik)

FORTRAN Ailesi (devam)

- FORTRAN 95: Pointer ve yapılara varsayılan olarak ilk değer atanması ve taşınabilirliğin mükemmel hale getirilmesi temel hedef olmuştur.
- Nesneye dayalı programlama özellikleri de FORTRAN dil ailesine sunulmaya çalışılmaktadır.

UYGULAMA ALANLARI

- Bilimsel Hesaplamalar, Mühendislik Uygulamaları ve Sayısal Analiz alanlarında özellikle FORTRAN 90 etkin bir biçimde kullanılmaktadır.

Functional Programming: LISP (1958)

- LISP Dili yapay Zeka uygulamaları için, MC Carthy ve ekibi tarafından 1958 yılında geliştirilmiştir.
- Atomlar ve Listeler adı verilen iki veri yapısına sahiptir.
- Program kodu ve veriler tam olarak aynı formdadır:
- Örnek: (A B C D)
 - Bu Eğer bir veri ise , A, B, C ve D verilerini içeren bir listedir
 - Eğer bir kod ise o zaman A fonksiyonunun B, C ve D parametrelerine uygulanması olarak yorumlanır.
- İki LISP versiyonu Scheme (1970) ve COMMON LISP (1984)

LISP (devam)

- LISP esas olarak yorumlayıcı kullanan bir dildir. Derleyici kullanan versiyonları da vardır.
- Veri tiplmesi bakımından esnek bir dildir.
- Olağanüstü esneklik, ifade gücü ve **kodun aynı zamanda veri olarak da kullanılabilmesi** özelliği LISP'i yapay zaka uygulamalarında rakipsiz hale getirmiştir.
- Nesne Yönelimli Programlamayı destekler
- Veri tabanlarına erişim ve GUI olanağı vardır.
- Çok iyi belgelendirilmiş olup, COMMON LISP, ANSI tarafından standart hale getirilmiştir.

ALGOL

- **ALGO**rithmic **L**anguage
- FORTRAN I'den esinlenilmiştir
- İlk kez 1958 yılında Avrupalı ve Amerikalı bir komisyonun Zürih'teki çalışmaları sonucu oluşturulan yüksek seviyeli bir dildir. İlk isim ALGOL 58 olarak verilmiştir.

ALGOL 60

- GAMM(German Society for Applied Mathematics and Mechanics) tarafından ALGOL'58'e eklemeler yapılmıştır.
- Amaçlar:
 - Matematiksel notasyonlara yakın ve kolay okunabilen
 - Literatürdeki hesap süreçlerinde kullanılabilen
 - Makine diline kolaylıkla çevrilebilen bir dil olsun
- Makineden bağımsız, daha esnek ve daha güçlübir dil olmuştur.
- Atama ifadesi olarak ilk evrensel kullanım bu dilde olmuştur.

variable := *expression*

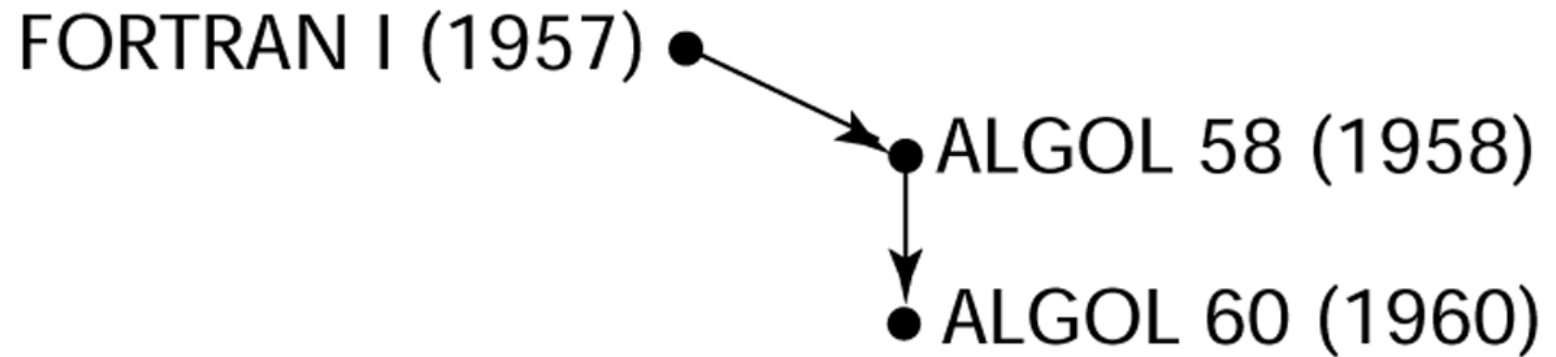
ALGOL 60: Başarılı yönler

- Blok yapısı kullanılmıştır
- Parametre aktarımı için, **pass by value ve pass by name** olmak üzere iki yöntem kullanılmıştır.
- **Rekürsiyon**'a izin verilmiştir.
- Stack-dynamic dizilere izin verilmiştir
- formal syntax tanımlı ilk dildir. Bu formal diller, Parsing teorisi ve derleyici tasarımı konuları için bir başlangıçtır.
- Rekürsiyon ve blok yapısını desteklemek için donanımda yığita izin verildiğinden bilgisayar mimarisini etkilemiştir.

ALGOL 60: Eksik yönler

- Anlaşılır olmakta zorluk ve yürütmede (implementation) verimsizliğe götürecek kadar esnek
- I/O ifadelerindeki yetersizlik yada zayıflıklar (eksiklik)
- 1960'lı yıllar için BNF(Backus Naur Form) kullanımı karmaşıklıklaştırmış gibi gözükmemektedir.
- Avrupalı bilim adamları arasında son derece popüler, eğitim ve araştırma aracı olarak kullanılır ve bilimsel makalelerde algoritmaları açıklamak için kullanılmasına rağmen;
- IBM tarafından desteklenmemiştir.
- Çünkü bu sırada IBM FORTRAN dilinde yazılmış zengin bir kütüphaneye sahiptir ve haliyle FORTRAN'ı desteklemektedir.
- O zamanlar IBM'in bilişim sektörünün %80'ine sahip olduğu düşünülürse, bu durum ALGOL60 için bir dezavantajdır.

ALGOL 60'ın şeceresi



COBOL: **C**ommon **B**usiness **O**riented **L**anguage) (1959-Mayıs)

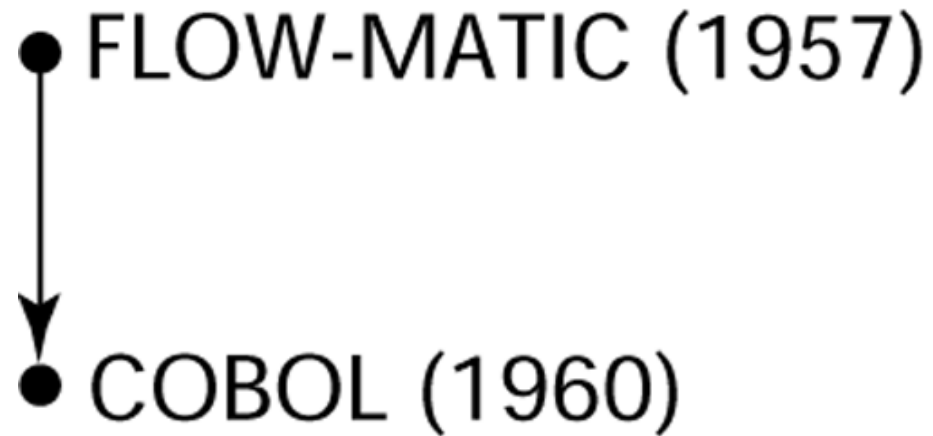
- Savunma bölümünün çalışmaları sonucu ortaya çıkmıştır.
- İngilizce sözcük yada cümle yapılarını kullanan ve İŞLETMELERe yönelik ortak bir dildir.
- Özellikle bilgisayara büyük miktarda bilgi giriş-çıkışının yapıldığı uygulamalar için geliştirilmiştir.(Stok kontrol, Bordro)
- Program içerisindeki dil ifadeleri ve fiziksel adresleri verilerin tanımlandığı ve çalıştırılabilir işlemlerin bulunduğu yer olarak ikiye ayrılır.
- Hiyerarşik veri yapıları ve yan anlamlı isimlerin (**connotative names**) görüldüğü ilk dillerdir.
- Fonksiyonları desteklememektedir.

COBOL (devam)

- COBOL içerisinde rapor yazdırma, tablo içinde arama yapma, ve kullanımı çok kolay olan dosya sıralama rutinleri bulunmaktadır.
- Yapısal özellikleri dolayısıyla veritabanı yönetim sistemleri ile birlikte kullanımı kolaydır.
- Tuğamiral Grace Murray Hopper, İlk önce yaklaşık 20 komuttan oluşan FLOW-MATIC'i geliştirdi ve COBOL'un temel modeli oldu.
- COBOL61-65-70-73, ANSI-COBOL versiyonları bulunmaktadır
- 1990'larda ise OOP versiyonu üretilmiştir.
- Üzerinde hala çalışmalar yapılmakta ve Amerika hala büyük miktardaki verileri işlerken COBOL kullanmaktadır.

Figure 2.4

COBOL'un seçeresi

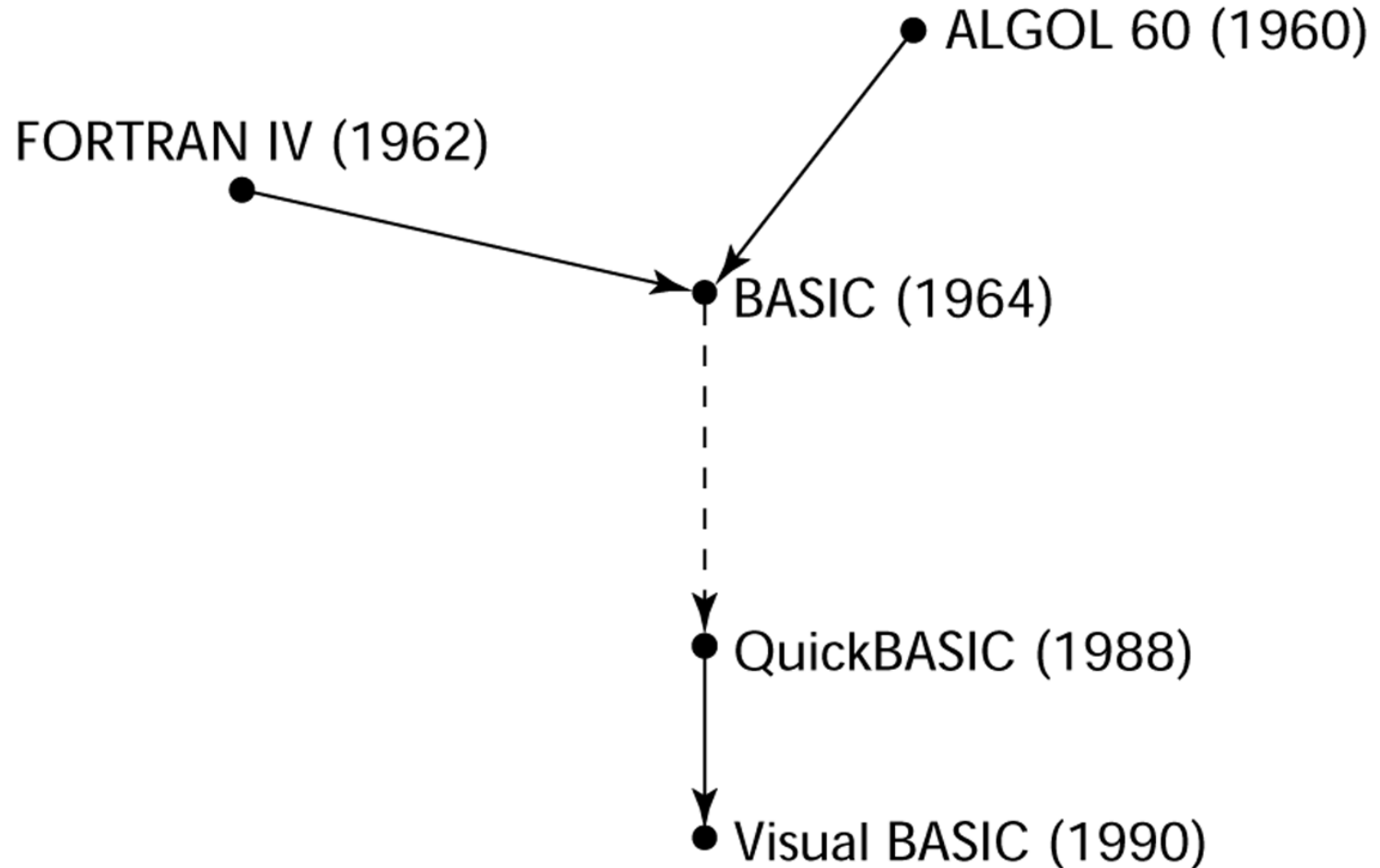


BASIC: İlk Zaman Paylaşımlı Bilgisayar Sistemi (1964)

- Beginner's All-purpose Symbolic Instruction Code
- Öğrencilerin bilgisayara daha kolay erişimlerini sağlamak ve basit ve etkin bir programlama dili ile program yazabilme isteklerine cevap vermek için tasarlanmış bir dildir.
- The first PL used through terminals connecting to a remote computer
- Sadece 14 komuta (LET, PRINT, GOTO...) sahipti. Tek veri tipi (number= kayan noktalı ve tamsayı)
- BASIC, FORTRAN ve ALGOL'den bazı bileşenleri almıştır. FORTRAN'dan DO çevrimini, ALGOL'den ise "until" yerine "to"

- Kolay bir dil ve genel maksatlı, belirli bir alana bağlı değil
- Uzman kişilere de hitap edebiliyor
- Açık ve anlaşılır hata mesajlarına sahip, kullanıcı bilgisayarla etkileşimli çalışabiliyor
- Küçük boyutlu programları hızlı bir biçimde çalıştırabiliyor
- Kullanım için donanım bilgisine sahip olmaya gerek yok
- Kullanıcıyı işletim sistemi ayrıntılarından dahi koruyabiliyor
- Derleyici kullanıyor, programın tümü makine diline çevrildikten sonra icra ediliyor
- BASIC'in pekçok versiyonları olmuştur. 1989'da ise nesne yönelimli uyarlama olan Visual BASIC ve 1998'de VB6.0 sunulmuştur.

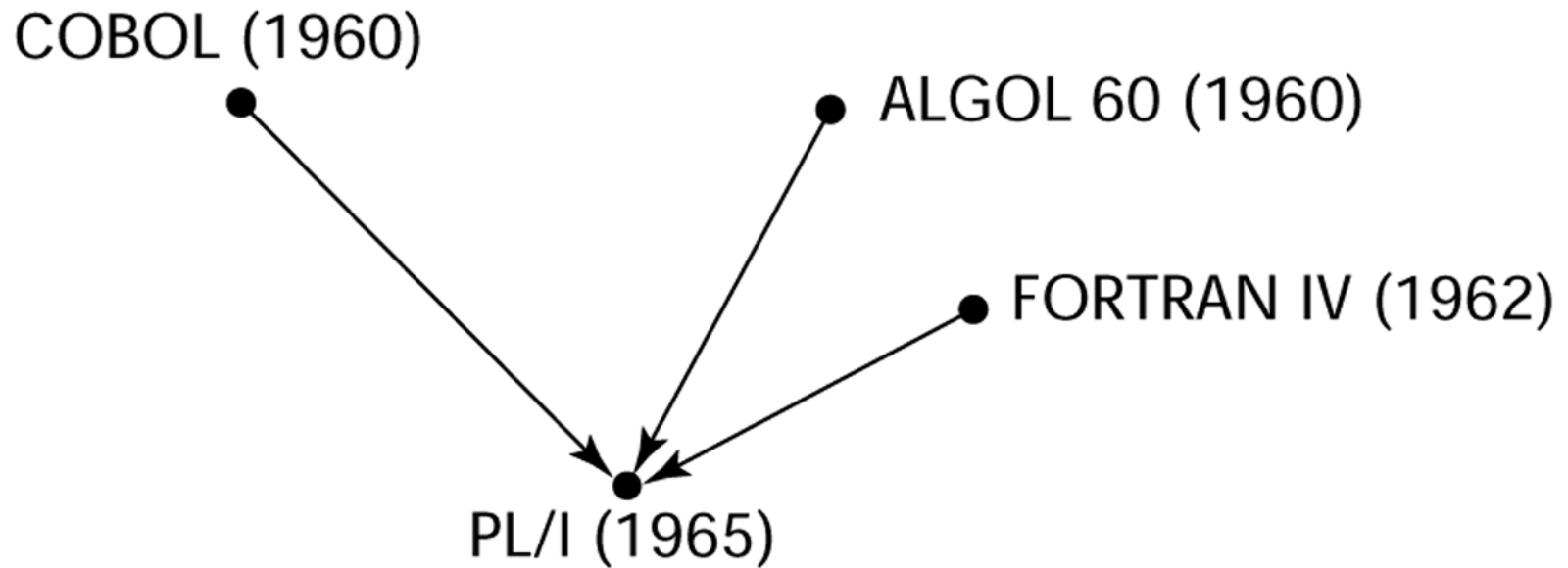
BASIC'ın Şeceresi



PL/1 (1965)

- COBOL, Fortran IV ve ALGOL 60'tan sonra bilimsel ve işletme problemlerine çözüm sağlayabilmek için floating-point ve desima veritiplerini desteklemek üzere geliştirilmiş bir dildir.
- IBM ve SHARE grubunun ortak çalışmalarının bir sonucu ortaya çıkan bir üründür.
- İlk PL :
 - Eşzamanlı olarak çalışabilen altprogramlara izin verebilmektedir.
 - run-time hatalarından başka 23 farklı tip.
 - Rekürsif olarak kullanılabilecek prosedürleri desteklemektedir.
 - Pointer kullanımına izin vermektedir.
 - Bir matrisin bir satırına vektor olarak işaret mümkün
 - Hafıza gereksinimi yüzünden karmaşık olabilmektedir.
- Tasarım yönünden iyi değildir.

Figure 2.6
Genealogy of PL/I



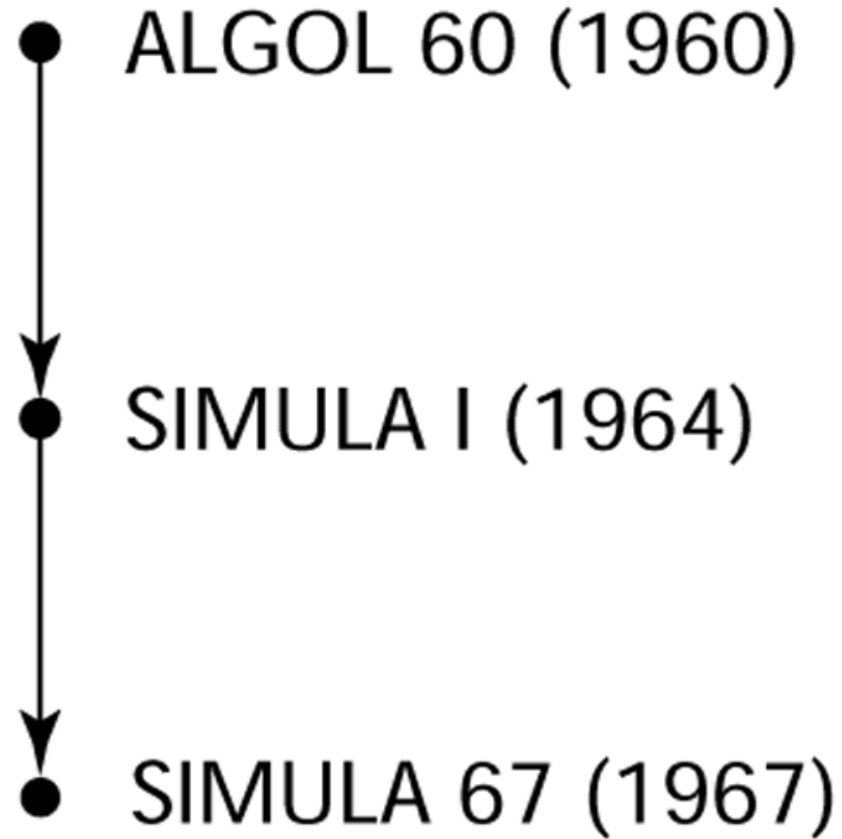
APL and SNOBOL: Dynamic Languages (1960s)

- APL (A Programming Language) dili tanımlayan bir bilgisayar mimarisi olarak IBM'de tasarlanmıştır.
- Dil ifadelerini oluşturan pek çok sayıda operatör vardır. Okuması oldukça zordur.
- SNOBOL (Snowball) metin işleyici (text processing) olarak Bell Lab'da özel olarak üretilmiştir.
- Yavaş bir dildir.
- Hem APL hem SNOBOL dinamik tiplleme ve dinamik bellek tahsisini desteklemektedir.
- SNOBOL dili yorumlayıcı kullanan bir dildir.
- Veri ve değişkenler üzerinde tip bakımından bir sınırlama bulunmuyordu
- Bu dilin devamı:awk, icon, perl,

SIMULA 67: Beginnings of Data Abstraction (1962 ve 1964 arasında)

- İlk olarak simülasyon için tasarlanmış bir dildir.
- ALGOL60'ın genişletilmiş versiyonudur. (Blok yapısı ve kontrol ifadeleri buradan alınmıştır)
- Daha önce durdurulduğu yerden itibaren yeniden çalışmaya başlayan altprogramları desteklemektedir.
- Veri soyutlamasına imkan veren sınıf yapılarını desteklemektedir

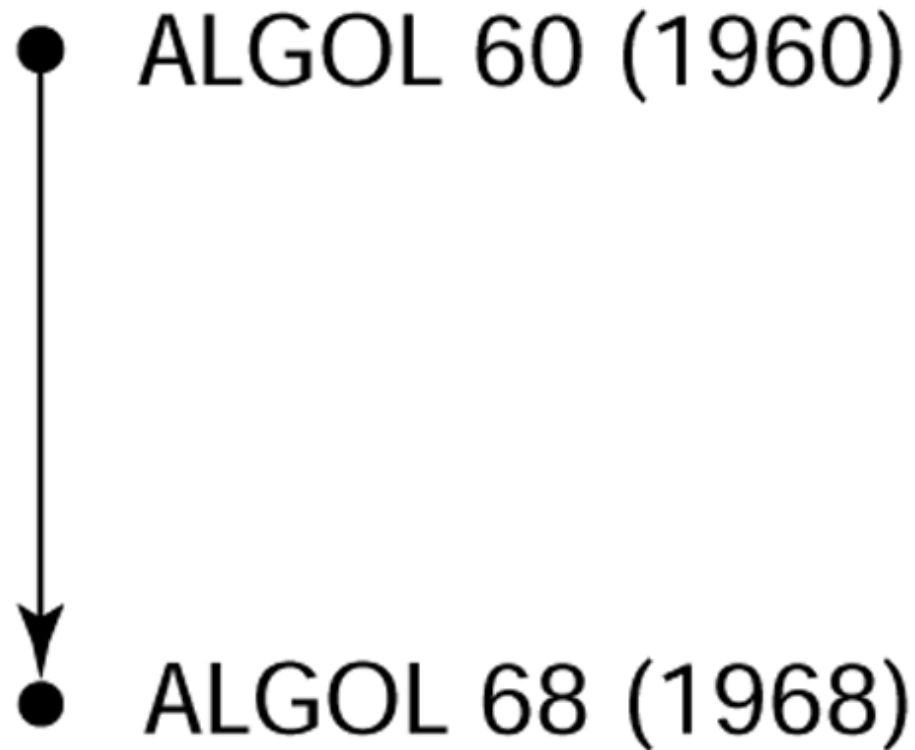
SIMULA 67'in Şeceresi



ALGOL 68

- Üstünlükler
 - Kullanıcı tanımlı(user-defined) veri tiplerini destekleyen ilk PL'dir.
 - ALGOL68'de flex adı verilen dinamik dizi kavramına izin veren ilk PL'dir.
- Eksiklikler
 - Öğrenilmesi oldukça zor olan bir gramer ve dil yapısına sahiptir.
 - Sadece bilimsel uygulamalar hedeflenerek tasarlanmış bir dil

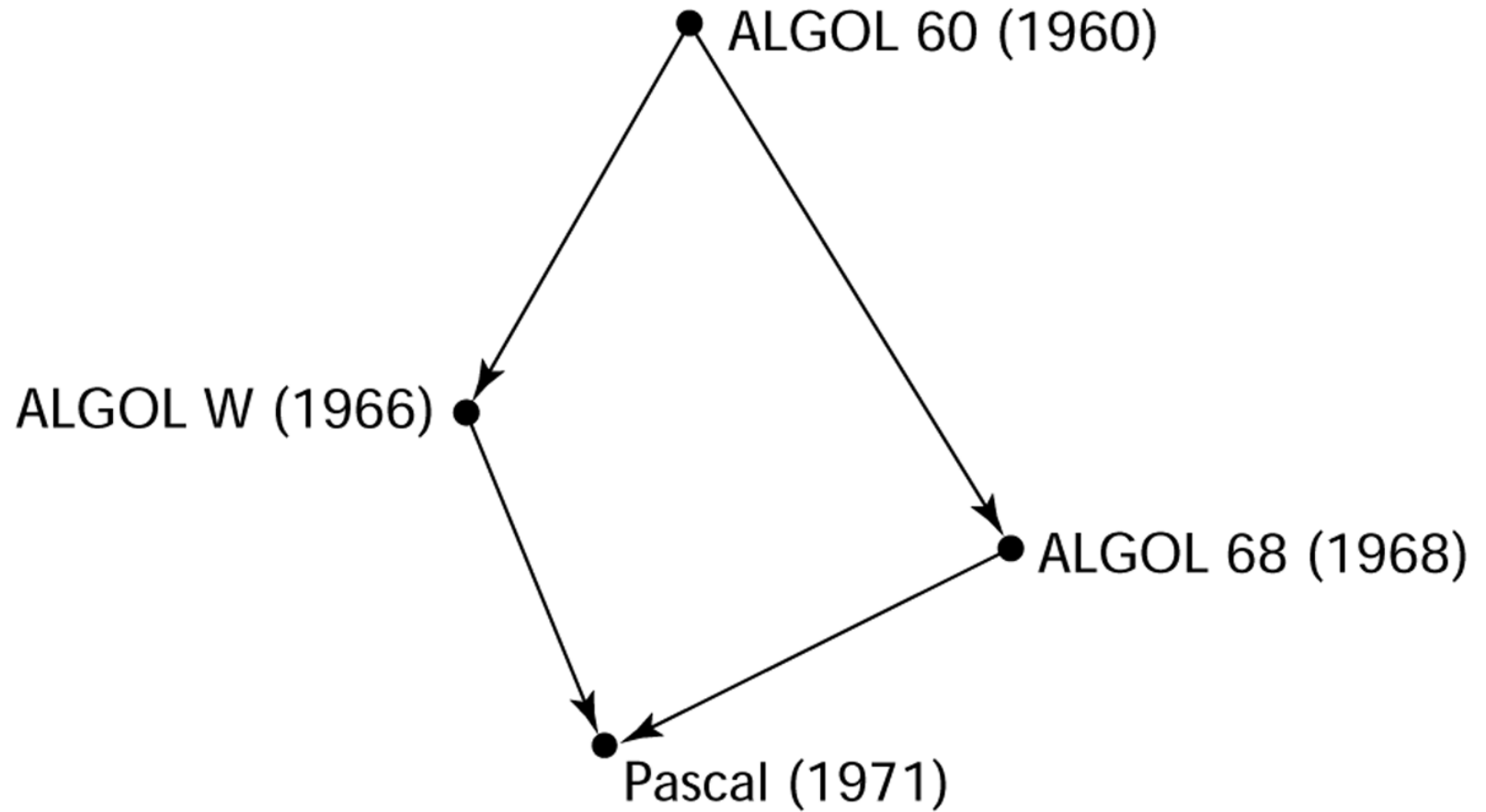
ALGOL 68'in şeceresi



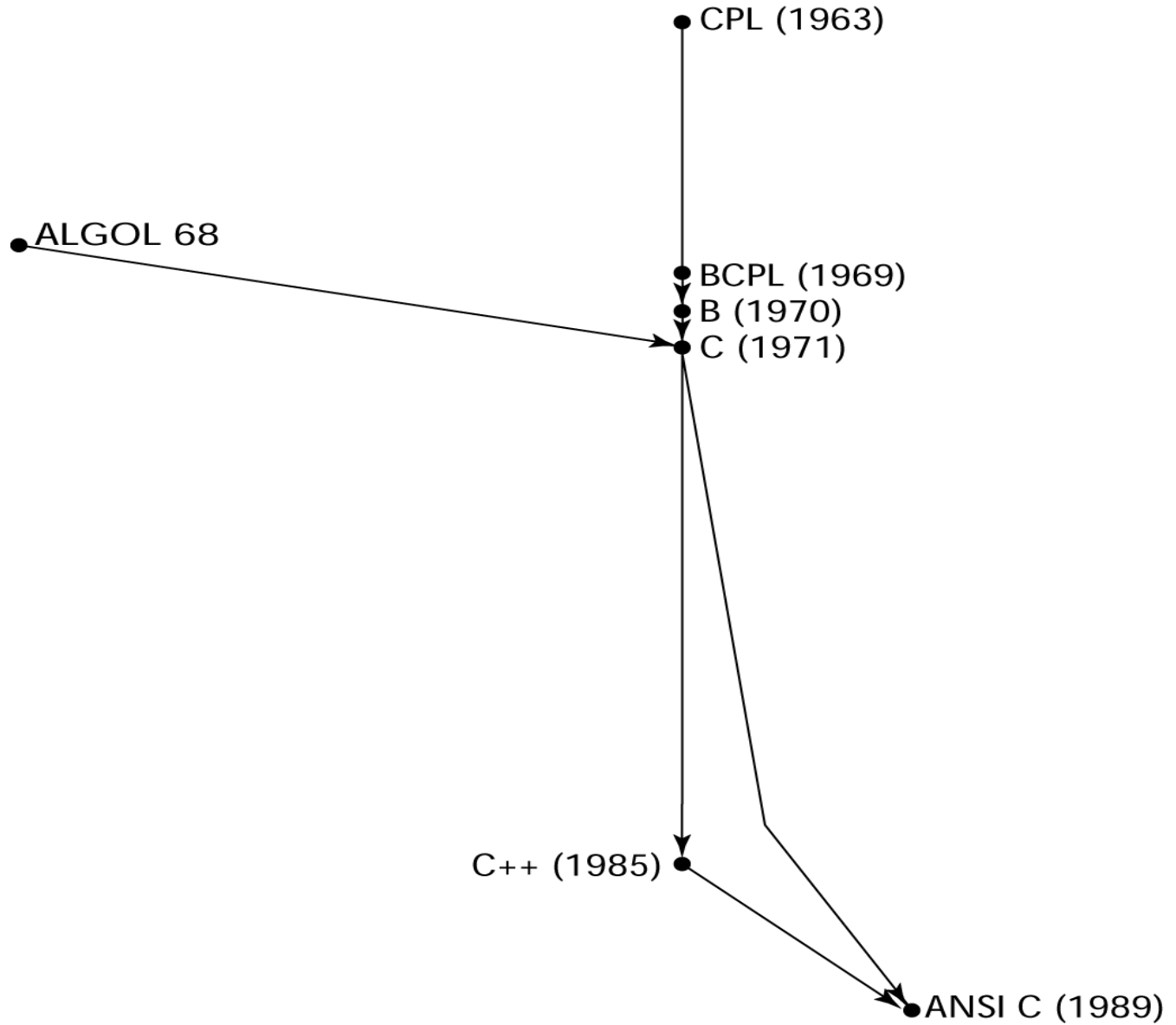
ALGOL'ün Torunları

- **Pascal (1971):**
 - İlk önceleri eğitim dili olarak kullanıldı
 - Basit ve okunabilir, yazılabilir (expressive)
 - Fortran and C ile karşılaştırıldığında emniyetli bir dildir.
- **C: Taşınabilir Sistem Programlama Dili (1972)**
 - Yeterli derecede kontrol ifadelerine ve veri yapısı olanaklarına sahip.
 - Tip kontrolü yetersiz.
 - Derleyicilerine çok kolay bir biçimde ulaşılabilir.

Pascal'in şeceresi



C'nin Şeceresi



ALGOL'ün torunları(devam)

- **Modula:**

- Yazılım Mühendisliği alanında önemli deneyimler sonucu ortaya çıkan ve güçlü bir tip ayrımı ve tip kontrolü mekanizmasına sahip bir dildir.
- Dinamik dizi kullanabilme (dinamik bellek yönetimi) ve eşzamanlılık özelliklerine sahiptir.
- Lilith adı verilen yeni bir bilgisayar sistemi için tasarlanmış fakat kullanılmamıştır.
- Modula 2 Pascal'a göre daha başarılı ve kullanıcıya esneklikler sunmakta. Modula 2'nin syntax'ı daha esnektir.
- Gerçek uygulama yazılım sistemleri için geliştirilmiştir.
- Soyut veri tipini destekliyor fakat miras alma (inheritance) özelliğini desteklemiyordu. Nesneye yönelim yok

ALGOL'ün torunları(devam)

- **Oberon:**

- Modula-2'ye dayanmakta fakat ondan daha basit idi.
- OOP'yi desteklemek için tiplere genişletilmişti. Tip ayrımı güçlüdür. OOP'i tam olarak destekler.
- Geleneksel veri türlerini ve kontrol yapılarının çoğunu içerir. Dinamik bellek yönetimi vardır.
- Endüstriyel gücü yüksek bir dil idi. Uygulama geliştirme amacıyla geliştirilmiş bir dildir.
- Veritabanı, ağ haberleşmesi, 3 boyutlu grafikler oluşturma ve nesne yönetimi için destek sağlayan zengin bir kütüphanesi vardır.
- Soyut veri türlerini destekler.

Prolog: Programming Based on Logic (1972)

- Çok üst düzey programlama dilinin ilk örneği
- PROgramming LOGic
- Mantık yürütme ve ispatlama tekniklerini kullanır
- PROLOG programı, bir nesneler kümesi ile bu nesnelerle ilişkili hedeflere nasıl erişilebileceğini tanımlayan kurallar kümesinden oluşur.
- Prosedürel bir yaklaşım değildir.
- Dil kurallar ve gerçeklerden oluşur.
- Verimsiz kalmıştır.
- Kullanım alanı belirli tipte VTYS ve yapay zeka alanlarıyla sınırlıdır.

ADA:

Tarihin en geniş tasarım çalışması

- ABD savunma bakanlığının bir çalışması sonucu ortaya çıkmıştır. Gömülü sistemlerin (embedded systems) programlanmasını sağlayacak PL üretimi amaçlanmıştır. (1983). Gerçek zamanlı uygulamalar için
- Augusta Ada Byron'ın (1815-1851) ölümünden sonra isimlendirilmiştir.
- Blok yapılı, nsne yönelimli, genel amaçlı ve eşzamanlılığı destekleyen bir dildir.
- Büyük boyutlu yazılımlar için uygundur.

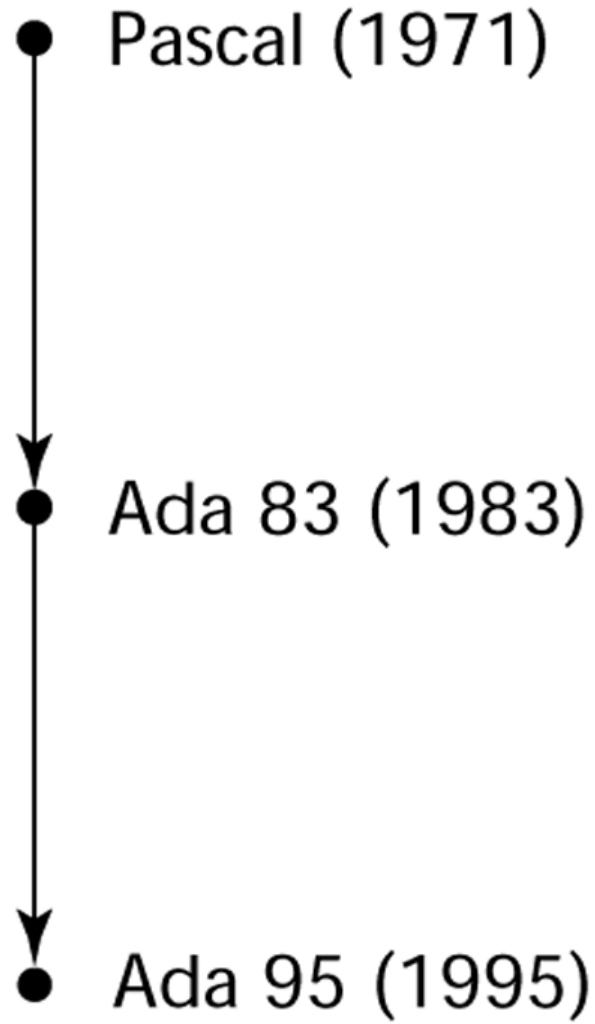
ADA (devam)

- Şablon (Generic) program birimleri sayesinde yazılımın yeniden kullanılabilmesini Buluşma yeri(rendezvous) mekanizmasının ilavesiyle eşzamanlı çalışmayı desteklemektedir. mechanism
- Çok geniş ve çok karmaşık bir dil. (Özellikle yazım kuralları)
- Çeşitli durumlara uygulanabilecek hazır şablonlara (template) sahiptir.
- İlk sıralar ADA derleyicileri kod üretmekte verimsiz idi.
- En çok gömülü sistemlerde başarılı olmuştur.
- Veri tipleri konusunda çok zengindir. Çok-iş işleme özelliğine sahiptir.

Ada 95

- ADA83'e kalıtım (inheritance), çok biçimlilik (polymorphism) gibi yeni eklemeler yapılarak ve tip türetim mekanizmasını genişleterek Ada 95 elde edilmiştir.
- Altprogramların dinamik kapsam bağlama kurallarına göre çağırılması mekanizması da ADA 95'in özelliklerine katılmıştır.

ADA'nın şeceresi

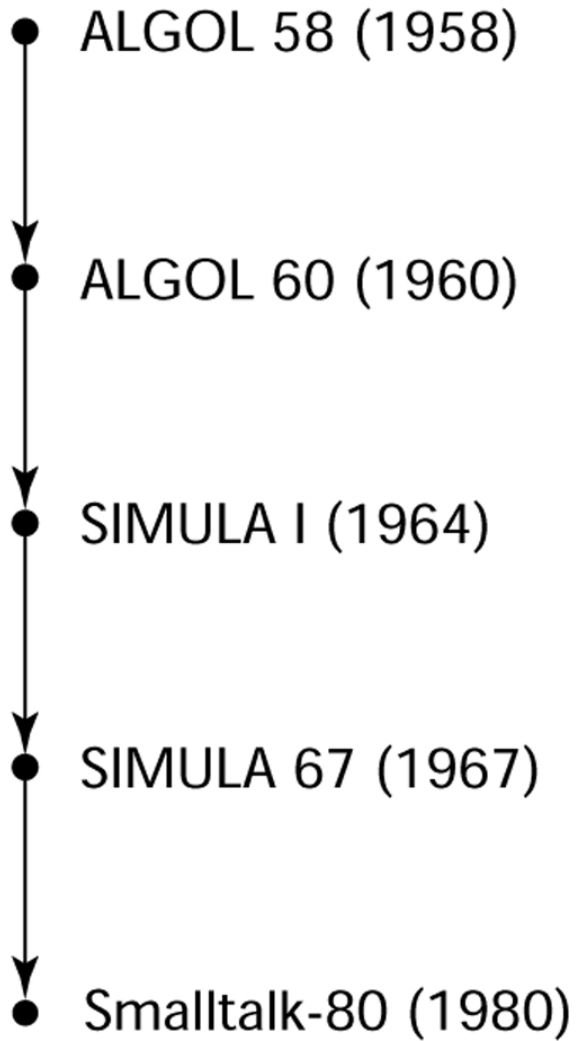


Smalltalk: Object-Oriented Programming

- OOP 3 temel karakteristiği: data abstraction, inheritance ve dynamic binding
- Smalltalk sadece bir PL değil, aynı zamanda yazılım geliştirme aracıdır.
- Program birimleri, verileri kapsülleyen nesneler ve yöntemlerden oluşur.
- Hesaplama bir nesneye bir mesaj göndererek yapılır

- İlk olarak Smalltalk-80 versiyonu Xerox dışında ticari amaçlı yazılımlarda kullanıldı. Halen yazılım geliştirmeleri için yeni teknolojileri de içererek kullanılmaktadır.
- **Smalltalk tamamen nesne yönelimli olan ve ticari ilk programlama dilidir.**
- Geliştirme ortamı nesne yönelimli bir alt yapısı olması nedeni ile çok gelişmiştir (tarayıcı, editör, debugger, açık kaynak kod).
- **Kaynak kodunun açık olması iyi bir eğitim ortamı haline gelmesini sağlamıştır. Diğer dillere göre çok basit bir sentaksı vardır ve Java gibi Geniş ve sürekli genişleyen bir programlama kütüphanesine sahiptir. Platform bağımsız bir dildir.**
- Bu dilde nesne yönelimli programlamanın 3 temel karakteristiği olan veri soyutlama (data abstraction), kalıtım (inheritance) ve dinamik bağlama(dynamic binding) kavramlarının hepsi bulunmaktadır.

Smalltalk'ın şeceresi



C++: (Imperative + OO)

- C ve Smalltalk'ın birleşimi
- Çoklu miras alma desteklenmekte
- Fonksiyon ve NYP 'ya izin verimektedir.
- Operatör ve metotların üstüste bindirilmesi desteklenmekte
- Sınıflar ve metotlar template edilebilir.
- Kuvvetli tip ayrımı, **dinamik bellek yönetimi**, hazır şablonlara sahip olma ve çok biçimlilik (polymorphism) özellikleri vardır.
- C'de bulunan özelliklerin çoğu burada vardır.
- PL/1 gibi geniş ve kompleks.
- Ada ve Java'dan daha az güvenli

Eiffel (1992)

- Imperative ve OO özellikleri birleştiren Hybrid PL
- Soyut veri yapılarını, kalıtımı ve dinamik bildirimleri destekler. (OO)
- Altprogram ve çağırıcı (caller) arasındaki iletişim için bildirimler (assertions) kullanır.
- C++'dan daha küçük ve basittir. Fakat ifade edilebilirliği ve yazılabilirliği neredeyse eşittir.

Delphi:

- Emir esaslı ve OO PL'nin başarılı bir biçimde birleştirilmesidir.
- Pascal'dan türemiştir. Bu yüzden, dizi elemanlarının kontrolünde, pointer aritmetiği ve tip zorlamalarında C ve C++'tan daha emniyetlidir.
- C++ 'dan daha az komplextir.
- Kullanıcı tanımlı opertaörlere, generic altprogramlara ve parametrize edilmiş sınıflara izin vermez.
- Daha iyi ve daha kolay yazılım geliştirtme için bir Graphical User Interface (GUI) sağlamaktadır.

Java: Emir Esaslı bir NDPD (OOL)

- Daha küçük, daha basit ve daha güvenilir bir PL tasarlamak üzere C++ tabanlı olarak geliştirilmiştir.
- Java, basit, taşınabilir ve nesneye yönelik özellikte bir dildir.
- Miras alma, çok biçimlilik, kuvvetli tip kontrolü, eş zamanlılık kontrolü, dinamik olarak yüklenebilen kütüphaneler, diziler, string işlemleri ve standart kütüphane gibi özellikleri vardır.
- Bir Java programının temel yapısal bileşeni sınıftır. Bütün veri ve metotlar bir sınıf ile ilişkilidir. Global veri yada fonksiyon yoktur.
- Hem referans değişkenleri ile hem de ilkel tiplerle erişilebilen sınıflara sahiptir.
- C++'ta bulunan çoklu miras alma, operatörlerin üstüste bindirilmesi, ve makro ön işemcisi özellikleri Java'da yoktur.
- Java'da şablon yapıları yoktur. İhtiyaç da minimuma inmiştir.

Java (devam)

- Pointer yoktur. Fakat bütün nesne sınıfları object adlı kök sınıftan miras almaktadır.
- Records, union or enumeration tipler yoktur.
- Prosedürel programlamayı desteklemez.
- Sadece tekli miras almayı destekler fakat çok gelişmiş bir GUI'ya sahiptir.
- İplik (**threads**) yapısına (**bir pencere içerisinde yeni pencereler açılmasına izin var**) sahip olduğundan eşzamanlılığı yönetmek kolaydır.
- **Çöp toplama (Garbage collection)** nesneler için belleği eniyi kullanımı sağlar.
- Tip dönüşümü kuvvetlidir.

Java (devam)

Java tipik olarak platformdan bağımsız olarak byte kodları biçiminde derlenir. Daha sonra bu byte kodlar bir Java görüntü makinesi adı verilen bir Java yorumlayıcısı tarafından kullanılacağı platformun makine dilindeki koduna çevrilir. Derlenmiş Java sınıflarının taşınabilirliğini garantileyen bir özellik vardır. Oda .class format adı verilen Java byte-kod dosya formatının kesin olarak tanımlanmış olmasıdır.

Java “applet” adı verilen başka sistemler içinde gömülü programların geliştirilmesi için de kullanılmaktadır. Bu appletler Internet Explorer yada Netscape gibi web tarayıcı programlar içerisinde kullanılabilir.

Java internet ve web programcılığında yaygın olarak kullanılmaktadır.

JavaScript: HTML-resident, client-side scripting language for the Web

- Netscape ve Sun Microsystems 'in ortak çalışmaları sonucu 1995'de üretilmiştir. Orijinal adı "LiveScript"tir.
- Dil bileşenlerinin içinde, web sayfalarının çeşitli kısımlarını ve özellikle HTML'i işleyecek ve kontrol edecek çeşitli olanaklar vardır.
- Javanın veri tipleri bakımından sert kısıtlamalarına karşılık, (Örn.Bütün değişken tipleri derleme zamanında belirlenir), Javascript bu açıdan toleranslıdır.(dinamik olarak tipleme)
- Java'nın HTML ile çalışma desteği zayıf iken, Javascript'in bu konudaki desteği tamdır.
- Javascript nesneye yönelik ve blok-yapısal özellikte bir dildir.

PHP: HTML-resident, server-side scripting language for the Web

- Personal Home Page (PHP), Rasmus Lerdorf tarafından 1994 yılında geliştirildi.
- Bir HTML dökümanı, okuyucu, hiper linkler boyunca diğer HTML dökümanlarına götürecek biçimde programlanabilir.
- Gömülü bir HTML belgesi browser tarafından istendiğinde PHP kodu Web server üzerinde yorumlanır.
- PHP'nin dizi yapısı JavaScript ve Perl dizi yapılarının bir birleşimidir.
- PHP ile form erişimi oldukça kolaydır.

C#

- C ve C++ dil ailesinin ilk bileşen yönelimli (Component-oriented) dilidir.
- C ve C++'dan derlenmiş, basit, modern, nesne yönelimli ve tür güvenli bir programlama dilidir.
- Yüksek başarımlı Common Language Runtime (CLR); bir yürütme motoru, bir çöp toplayıcı (garbage collection), anında derleme, bir güvenlik sistemi ve zengin bir sınıf çerçevesi (.NET Framework) içerir. CLR temelden, birden çok dil desteğine kadar herşey için tasarlanmıştır.
- CLR'ı hedef alan diller: Visual C#, Visual BASIC .NET, Managed C++, J#.NET ve Jscript.NET
- Common Language Specification, CLS, dil işlevselliğinin yaygın bir düzeyini tanımlar. .NET Framework işlevselliğine tam erişim ve ve diğer uyumlu dillerle zengin birlikte çalışabilirlik vardır.

C# (devam)

- C# otomatik bellek yönetimini kullanır. C# tür sistemi işaretçi türleri ve nesne adreslerinin doğrudan değiştirilmesine de izin verilir.
- C# tür sistemi bileşiktir. Her şey bir nesnedir. Kutulama ve kutuyu açma gibi kavramların yenilikçi kullanımı ile C#, her veri parçasının bir nesne olarak değerlendirilmesine olanak sağlayarak, değer türleri (value type) ve başvuru türleri (reference type) arasındaki açığı kapatır.