

Bölüm: 0

- 0.1. Giriş ve Amaçlar.
- 0.2. Programlama Dili Nedir?
- 0.3. Programlama Dillerini Düzeylere Ayırmak.
- 0.4. Dil Çevrimi.
- 0.5. Programlama Dillerinin Yazılım Yaşam Döngüsündeki Yeri.
- 2. Programlama dili:
 - . Makine dili:
 - 4. . Yorumlayıcılar (Interpreters).
 - . Derleyiciler (Compiler).
 - . Metinsel (Lexical) Analiz. (Sözcüksel).
 - . Sözdizim (Syntax) Çözümleme.
 - Ayırıştırma.
 - Belirsizlik.
 - . Anlam (Semantics) Çözümleme.
 - Anlam Çözümleme.
 - Soyut dil.
 - Kod oluşturma.
 - Optimizasyon (Eniyileme).
 - . Dil çevrim Yöntemlerinin Karşılaştırılması.
 - Zaman.
 - Bellek kullanma.
 - Hata Bildirme.
 - Taşınabilir Kod.
- 5. . Gereklerin Analizi ve Belirlenmesi.
 - . Sistem Tasarımı.
 - . Gerçekleştirme ve Birim Sınama.
 - . Sınama.
 - . Bakım.

Bölüm: 1

- 1.1. Programlama Dilleri Kavramlarını Öğrenmenin Nedenleri.
 - . Fikirleri ifade etme yeteneğinin artırılması.
 - . Uygun dilleri seçebilme bilincinin geliştirilmesi, altyapının sağlamlaştırılması.
 - . Yeni diller öğrenme yeteneğinin geliştirilmesi.
 - . Implementasyonun (Gerçekleştirim) öneminin daha iyi anlaşılması ve kavranması.
 - . Bilinen programlama dillerinin daha iyi kullanılması.
 - . Bilgisayar biliminde kapsamlı ilerleme.
- 1.2. Programlama Alanları.
 - . Bilimsel Uygulamaları: Fortran.
 - . Ticari Uygulamaları: Cobol.
 - . Yapay Zeka: Lisp, Prolog.
 - . Sistem Programlama: C.
 - . Web Yazılımı:
 - markup: XHTML.
 - scripting: php.
 - Genel amaçlı: Java.
- 1.3. Dil Değerlendirme Kriterleri.
 - . Okunabilirlik (Readability).
 - . Yazılabilirlik (Writability).
 - . Güvenilirlik (Reliability).
 - . Maliyet (Cost).
 - . Diğerleri:
 - Taşınabilirlik (Portability).
 - Genellik (Generality).
 - İyi tanımlanmış olması (Well-definedness).

- Okunabilirlik:
 - . Genel Kolaylık.
 - . Ortogonalite.
 - . Kontrol ifadeleri.
 - . Veri tipleri ve Veri Yapıları.
 - . Syntax Tasarımı.
- Yazılabilirlik:
 - . Basitlik ve Ortogonalite.
 - . Soyutlama (Abstraction) Desteği.
 - . Anlamlılık (Açıklayıcılık) (Expressivity).
- Güvenilirlik:
 - . Tip kontrolü (Type checking).
 - . İstisna işleme (Exception Handling).
 - . Farklı Adlandırma (Aliasing).
 - . Okunabilirlik ve Yazılabilirlik.
- Maliyet:
 - . Dili kullanacak programcıların eğitilmesi.
 - . Programları Yazma (özel uygulamalara yakınlık).
 - . Programları derleme (Compiling) maliyeti.
 - . Programları yürütme (Executing) maliyeti.
 - . Dil implementasyon sistemi.

- 1.4. Dil Tasarımını Etkileyen Faktörler.
 - . Bilgisayar Mimarisi (Computer Architecture).
 - . Programlama Tasarım Yöntemleri.
- 1.5. Dil Kategorileri. (Programlama Paradigmaları)
 - . Zorunlu (Imperative):
 - Temel özellikleri değişkenler (variables), atama ifadeleri (assignment statements) ve iterasyondur (iteration).
 - C, Fortran, PL/I, Pascal, Cobol, Ada.
 - . Fonksiyonel (Functional):
 - Hesaplamaları yapmanın temel yolu, fonksiyonları verilen parametrelere uygulamaktır.
 - Lisp, Scheme, ML.
 - . Mantıksal (Logic):
 - kural tabanlı (Rule-based) (kuralların sırası belirli değildir).
 - Prolog.
 - . Nesneye - dayalı (Object-Oriented):
 - Veri soyutlama (Data abstraction), miras (inheritance), geç bağlama (late binding).
 - Simula 67, C++, Java, Smalltalk.
 - . İşaretleme (etiketleme) (Markup):
 - Yeni; kendi başına programlama değildir, fakat web dokümanlarında bilgiyi düzenlemek için kullanılır.
 - XML, XHTML.
 - . Diğerleri:
 - Visual (görsel) diller: Visual Basic, Net.
 - Scripting diller: JavaScripting, Ruby, Perl.
 - Hibrid diller (Markup, Programlama dili).
 - özel amaçlı diller.
- 1.6. Dil Tasarımında Verilen Ödünler.
 - . Güvenilirlik ve Yürütme (Execution) maliyeti.
 - . Okunabilirlik ve Yazılabilirlik.
 - . Yazılabilirlik ve Güvenilirlik.
- 1.7. Implementasyon Yöntemleri.
 - . Derleme (Compilation).
 - . Saf Yorumlama (Pure Interpretation).
 - . Hibrit (Melez-Hybrid) implementasyon sistemleri.
- 1.8. Programlama Platformları.
 - . Borland JBuilder.
 - . Microsoft VS.NET.
 - . NetBeans.

• Bölüm 2.

- Eniac : 1943.

2.1. Zuse'nin Plankalkül'ü.

- ilk yüksek seviye p. d.
- Asla geliştirilmedi.
- ileri veri yapıları (kaynak nokta, diziler, kayıtlar).
- sabitler.

2.2. Minimum Donanım Programlama: Sözde Kodlar.

- Short Code (Kısa Kod) 1949. by Mauchly.
- Speedcoding (Hızlı Kodlama) 1954. by Backus → IBM701

2.3. IBM704 ve Fortran.

- Fortran 0: 1954 ... IBM → uygulanmamıştır.
- Fortran 1: 1957: Ticari olarak kullanıma sunuldu.
- Fortran 2: 1958: • Bağımsız derleme.
• Hataları (bugs) düzelti.
- Fortran 4: 60-62: → Belirtilmiş tip tanımlamaları.
→ Mantıksal seçim ifadesi.
→ Altprogram isimleri parametre alabilir.
- Fortran 77: 1978: • karakter dizisi işleme.
• Mantıksal dögü kontrol ifadesi.
• if-then-else ifadesi.
- Fortran 90: 1990: • Modüller.
• Dinamik diziler.
• İşaretçiler (pointers), bağılı liste, dinamik bellek.
• Özyineleme (Recursion).
• Case ifadesi.
• Parametre tipi testi.
• Bit düzeyinde işlem.
- Fortran 95.
- Fortran 2003.

2.4. Fonksiyonel Programlama: Lisp. by McCarty → IBM704.

- AI araştırmasının ihtiyaç duyduğu dil şöyleydi:
• Veriyi liste halinde işleme. (dizi yerine)
• Sembolik hesaplama. (sayısal yerine).
- Sadece iki veri tipi: atomlar ve listler.
- Sentaks lambda calculus'a (λ) dayalıdır.
- Lisp esas olarak yorumlayıcı kullanan bir dildir. Derleyici kullanan versiyonları da vardır.
- Nesne Yönelimi programlamayı destekler.
- Common Lisp ve Scheme, Lisp'in güncel diyalektleridir.
- ML, Miranda, ve Haskell ilgili dillerdir.

2.5. Sofistikeliğe doğru ilk adım: Algol 60.

- Fortran 1'den esinlenilmiştir.
- ilk makine-bağımsız dildir.
- Sentaksı resmi olarak tanımlanan ilk dildir (BNF).

2.6. Ticari Kayıtları bilgisayara uyarlamak: Cobol.

- Bir yüksek-düzeyli dilde ilk kez makro olanağı.
- Fonksiyonları desteklememektedir.
- 1990'larda ise OOP versiyonu üretilmiştir.

2.7. Zaman Paylaşımının (TimeSharing) başlangıcı: Basic.

- kullanıcının zamanı bilgisayarın zamanından değerlidir.
- Mevcut popüler diyalekt: Visual Basic.
- Sadece 14 komuta sahiptir. Tek veri tipi (number).
- Basic, Fortran ve Algol'den bazı bileşenleri almıştır.
- Derleyici kullanıyor.

2.8. Herkes için Herşey: PL/1. (by IBM, Share).

- En önemli özelliği çeşitli uygulama alanları için kullanılabilmesidir.
- Pointer kullanımına izin vermektedir.
- Rekürsif olarak kullanılabilecek prosedürleri desteklemektedir.
- PL/1 Katkıları: - unit level concurrency.
- exception handling.
- Switch Selectable, Recursion.
- pointer.
- array cross sections.

2.9. İlk iki Dinamik Dil: APL ve Snobol.

- Dinamik tip belirleme ve dinamik bellek ayrımı ile karakterize edilir.
- Değişkenlerin tipi yoktur.
- APL: 1960'larda IBM'de Ken Iverson tarafından bir donanım tanımlama dili olarak tasarlanmıştır.
- Snobol: Bell Lab'da Farber, Griswold ve Polensky tarafından string ve text işleme dili olarak tasarlanmıştır.
• Yorumlayıcı kullanan bir dildir.

2.10. Veri Soyutlama (Data Abstraction) nın başlangıçları: Simula 67.

- Asıl olarak sistem simülasyonu için tasarlanmıştır.
- Algol 60 ve Simula 1'e dayalıdır.
- Nesneler ve miraslar.

2.11. Ortogonal Dizayn: Algol 68.

- kullanıcı-tanımlı veri yapıları destekleyen ilk dil.
- Dinamik dizilere izin veren ilk dil.

2.12. Algol'lerin ilk torunlarından bazıları:

- Algol dilleri bütün zorunlu (imperative) dilleri etkiledi
→ pascal, C, Modula, Ada, oberon, C++/Java, Perl.
- pascal: Yapısal Programlama (structured) öğretmek için tasarlandı.
- C: Sistem Programlama için tasarlandı.
• Esnek Yapı, güçlü operatörler, fakat zayıf tip kontrolü.

- Modula: Gerçek uygulama yazılım sistemleri için geliştirilmiştir.
• Soyut veri tipini destekliyor fakat miras, Nesneye yön. yok.

2.13. Mantık temelli Programlama: Prolog.

- Mantık yürütme ve ispatlama tekniklerini kullanır.
- Prosedürel değildir.
- Dil kuralları ve gerçeklerden oluşur.
- Temel uygulama alanı: Doğal dil işlemedir.

2.14. Tarihin en büyük tasarım çabası: Ada.

- DoD tarafından gömülü (Embedded) sistemler için geliştirilmiştir.
- Veri tipleri konusunda çok zengindir. Çok iş işleme özelliğine sahiptir.

2.15. Nesneye dayalı Programlama: Smalltalk.

- Bir N-D dilin ilk tamamen implementasyonu (veri soyutlama, miras, dinamik tip bağlama).
- GUI tasarımına öncülük etmiştir.
- OOP'yi yükseltmiştir.
- Smalltalk tamamen NY olan ve ticari ilk programlama dildir.

2.16. Zorunlu (Imperative) ve ND(OO) özellikleri birleştirmek: C++.

- C ve Simula 67 den geliştirilmiştir.
- Hem prosedürel (fon. izin verir) hem de OO programlamayı desteklediği için geniş ve karmaşık bir dildir.
- Çoklu miras alma desteklenmekte.

2.17. Bir zorunlu ND dil (Imperative-Based OO): Java.

- C++ temellidir.
- önemli derecede basitleştirilmiştir.
- Sadece OOP yi destekler.
- Referansları vardır, pointer yoktur.
- Java'applet' adı verilen başka sistemler içinde gömülü programların geliştirilmesi için de kullanılmaktadır.

2.18. Ağ (web) için Betik Diller (Scripting Languages):

- Perl, JavaScript: tamamen yorumlayıcıdır.
- php: web uygulamaları için kullanılır, tamamen yorumlayıcıdır.
- python: Nesne tabanlı yorumlayıcıya sahip bir metin dildir.
- Ruby: Tüm veriler nesnedir.

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

- Bütün .Net dilleri ortak tip sistemi (CTS) kullanır, bu ortak bir sınıf kütüphanesi sağlar.

2.20. İşaretleme (Markup) / Programlama Hibrit Diller:

XSLT, JSP.