Programlama Dillerinin Prensipleri

Hafta 14 – Fonksiyonel Programlama

Dr. Öğr. Üyesi M. Fatih ADAK

İçerik

- Tarihsel gelişim
- Tanım
- Neden fonksiyonel paradigma?
- Yaklaşım
- Soy ağacı
- Fonksiyonel dillerin yapısı
- Değişkenin rolü
- Haskell dili
- Lisp dili
  - Formlar
  - Veri türleri
- Fonksiyonel ile Emir Esaslı karşılaştırılması

Tarihsel Gelişim

- Fonksiyonel tasarım ilk John McCarty tarafından 1956 yılında tanıtılmıştır.
- En güçlü temsilcisi Lisp dilidir. Bu isim güçlü liste işlemleri yapabilmesinden gelir.

John McCarthy (1927 – 2011)

Tanım

- Fonksiyonel dillerin tasarımı Matematiksel Fonksiyonlara dayalıdır ve değişkenler(variables), matematikte olduğu gibi gerekli değildir.
- Kullanıcıya yakın olan sağlam bir teorik temele sahiptir.
- Fonksiyonel programlamada , bir fonksiyon aynı parametreler verildiğinde daima aynı sonucu üretir (referential transparency).

Neden Fonksiyonel Paradigma?

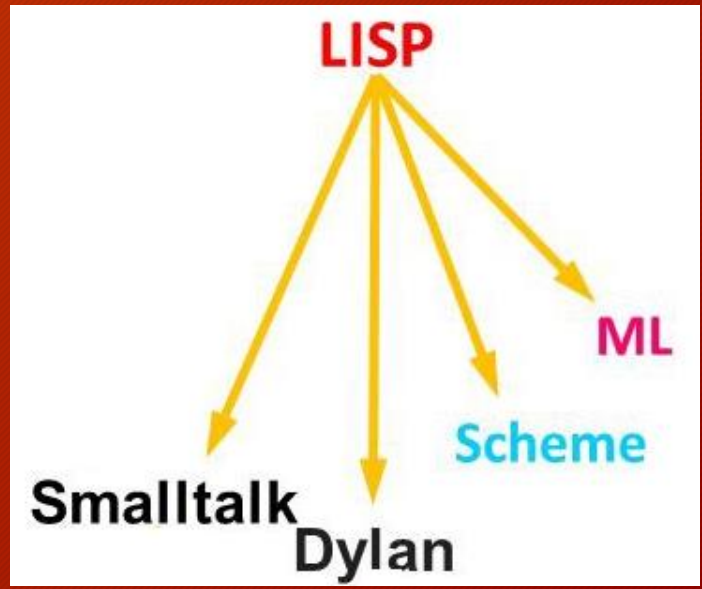
- Tarihsel süreçte emir esaslı tasarımdan sonra tanıtılmıştır.
- Emir esaslı dillerin tasarımı doğrudan doğruya von Neumann mimarisine dayanır.
- Bir emir esaslı dilde, işlemler yapılır ve sonuçlar daha sonra kullanım için değişkenlerde(variables) tutulur. Emir esaslı dillerde değişkenlerin yönetimi karmaşıklığa yol açar.

Yaklaşım

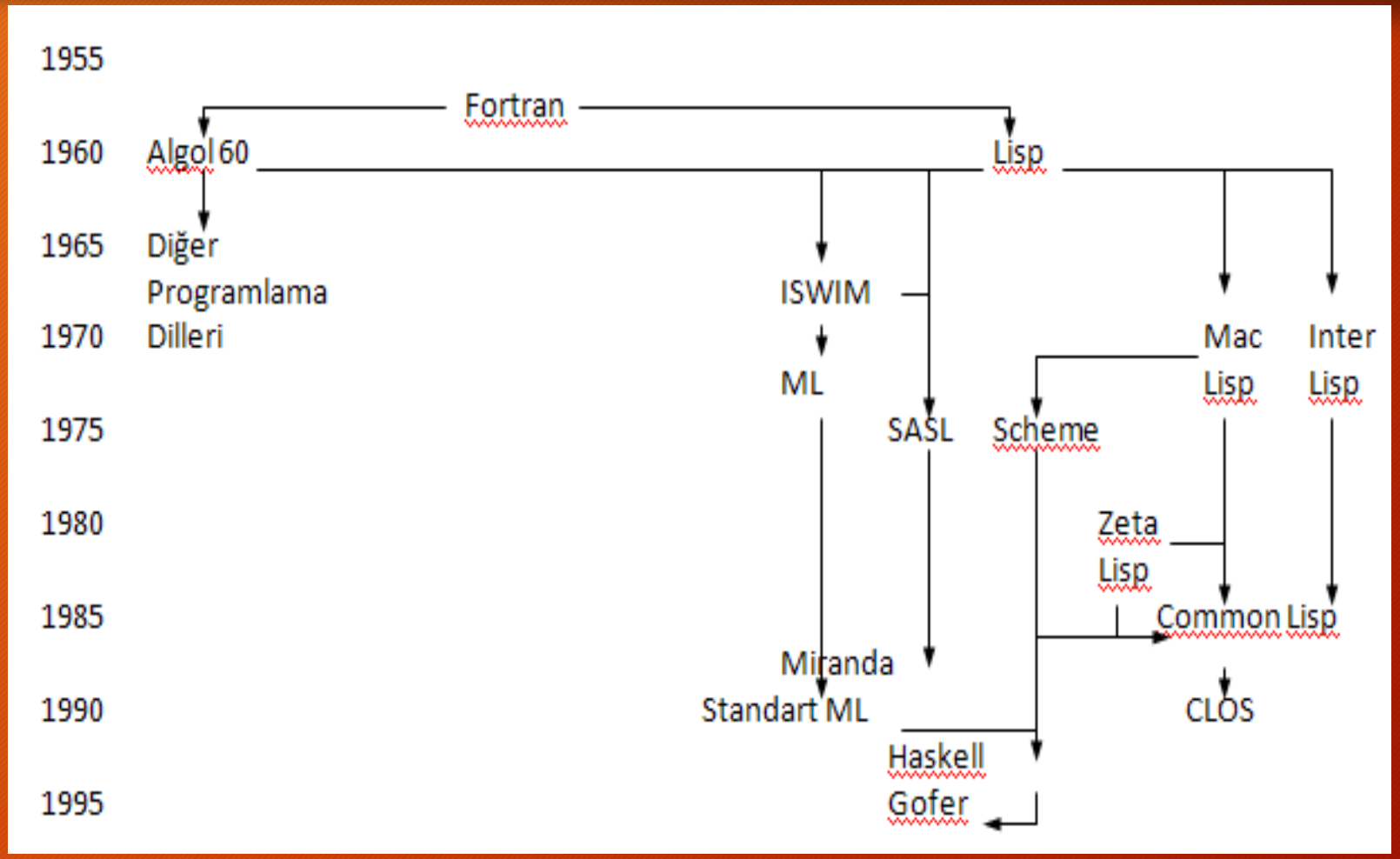
- Fonksiyonel dillerde problemin nasıl çözüleceğinden çok problemin ne olduğu önemlidir.
- For, if, while gibi denetim mekanizmaları makrolar halinde sunulur ve özyineleme ile gerçekleştirilir. Daha çok yapay zeka ve benzetim uygulamaları için uygun olabilir.
- Fonksiyon yaklaşımından dolayı matematik temeli oldukça sağlam olacağından optimize edilme (en iyileme) şansı çok yüksektir.

Yaklaşım devam…

- Fonksiyonel programlama paradigması, Programlama dilini fonksiyon tanımının temel biçimleri üzerine oturtarak, algoritmaların ifadesi için basit ve açık bir ortam elde etmeyi amaçlamıştır.
- İlk örnek LISP dilidir ve onu ML, Scheme ve Haskell, dilleri izlemiştir.



Soy Ağacı


1955
Fortran
1960
Algol 60
Lisp
1965
Diğer
Programlama
1970
Dilleri
ISWIM
ML
Mac
Lisp
Inter
Lisp
1975
SASL
Scheme
1980
Zeta
Lisp
1985
Common Lisp
Miranda
1990
Standart ML
CLOS
Haskell
1995
Gofer

Fonksiyonel Dillerin Yapısı

- Sadece fonksiyonlar üzerine kurulmuş bir modeldir.
- Fonksiyonlar bir çok değer alır ve geriye sadece bir değer döndürürler.
- Fonksiyonlar başka fonksiyonları çağırır ya da başka fonksiyonun parametresi olurlar.
  Fonskiyon(..(fonksiyon2(fonksiyon1(veriler)))..)
- Bu dillerde, alt yordamlar,fonksiyonlar (prosedürler) kullanılarak program daha alt parçalara bölünür.

Fonksiyonel Dillerin Yapısı devam…

- Fonksiyonel diller Sembolik veri işleme amacı ile dizayn edilmiştir.
- Bu diller;
  - Türev ve integral hesaplamalarındaki
  - Elektrik devre teorisideki
  - Matematiksel mantık oyunlarındaki
  - Yapay zekanın diğer alanlarındaki

sembolik hesaplamalarda kullanılmaktadır.

- Karmaşık hesaplamalar daha basit ifadeler cinsinden yazılarak kolaylıkla çözümlenebilir.

Değişkenin Rolü

- Fonksiyonel olmayan tasarımlarda değişken, bir değeri tutan yer rolünü üstlenirken fonksiyonel tasarımda direk değerin kendisidir.

x = x + 1 ifadesinde her x farklı bir değeri temsil eder.

10 = 9 + 1 deki gibi düşünülebilir.

Haskell Dili

- Bağımlı ve bağımsız değişkenlerin tespit edilip hangi işlerin eş zamanlı çalıştırılabileceği belirlenir.
- Tam olarak fonksiyonel bir dildir. (değişkenler yoktur, atama ifadeleri yoktur, hiçbir çeşit yan etki yoktur).
- Tembel değerlendirme(lazy evaluation) kullanır (değer gerekmediği sürece hiçbir alt-ifadeyi değerlendirme)
- Liste kapsamları(list comprehensions), sonsuz listelerle çalışabilmeye izin verir.

Lisp Dili Formları

- ANSI Common Lisp (cLisp)
  - Derleyici, yorumlayıcı, debugger içerir
- GNU Common Lisp (gcl)
  - Derleyici, yorumlayıcı içerir
- Allegro CL (Commercial Common Lisp Implementation)

Lisp Dili Veri Türleri

- İki ana veri türünden oluşur.
  - Atom ve List
- Atom Veri Türü
  - String
  - Tam ve Ondalık sayılar
  - Karmaşık sayılar

Fonksiyonel ile Emir Esaslı Tasarım Karşılaştırması

Emir Esaslı (imperative) diller	Fonksiyonel diller
Verimli çalışma	Verimsiz çalışma
Karmaşık semantik	Basit semantik
Karmaşık sentaks	Basit sentaks
Eş Zamanlılık (kullanıcı tanımlı)	Eş Zamanlılık (Otomatik)

Kaynaklar

- Yumusak N., Adak M.F. *Programlama Dillerinin Prensipleri*. 1. Baskı, Seçkin Yayıncılık, 2018
- Sebesta, Robert W. *Concepts of programming languages*.11 ed. Pearson Education Limited, 2016.
- Sethi, Ravi. *Programming languages: concepts and constructs*. Addison Wesley Longman Publishing Co., Inc., 1996.
- Watt, David A. *Programming language design concepts*. John Wiley & Sons, 2004.
- Malik, D. S., and Robert Burton. *Java programming: guided learning with early objects*. Course Technology Press, 2008.
- Waite, Mitchell, Stephen Prata, and Donald Martin. *C primer plus*. Sams, 1987.
- Hennessey, Wade L. *Common Lisp*. McGraw-Hill, Inc., 1989.
- Liang, Y. Daniel. Introduction to Java programming: brief version. pearson prentice hall, 2009.
- Yumusak N., Adak M.F. *C/C++ ile Veri Yapıları ve Çözümlü Uygulamalar*. 2. Baskı, Seçkin Yayıncılık, 2016