PROGRAMLAMA VE PARADİKMALARI

1.BÖLÜM: Giriş

Programlama paradikmalarının gelişimi ve tarihi:

Temelleri 1.dünya savaşı sonrasında ortaya çıkan teknonojik ve yazılımsal alanlardaki ilerlemeler sonucu ortaya çıkmış ve bir disiplin halini almaya başlamıştır. Temelleri Amerika Birleşik Devletleri, İngiltere ve Almanya’da atılmıştır. Ve özellikle Almanya’da ortaya çıkan ve temelinde karşılıklı şifreleme ve haberleşme olan “elektromekanik“ bir yapısı olan “Enigma“ makinaları ile atılmıştır.

2.Dünya savaşı sırasında ortaya çıkan ve temelinde ortaya çıkan ve temelinde Alman gizli haberleşme sistemlerini ortaya çıkarmak üzere Alan Turning ve arkadaşlarının ortaya çıkarmış olduğu matematiksel ve mantıksal yaklaşımlar bu disiplinin ortaya çıkmasına yardımcı olmuştur. Yine 1944 yılında ortaya çıkan ve dünyanın ilk modern bilgisayar yapısı olarak kabul edilen “ENIAC” bu disiplin için önemli bir kilometre taşı olmuştur.1960’lı yıllarda ortaya çıkan ve yapay zeka uygulamalarının atası olarak kabul edilen “LOGİC THEROİES” ile yeni bir boyut kazanmıştır.

2.BÖLÜM: Neden ortaya çıktı

Gelişen teknonoji ve artan yeni ihtiyaçlar nedeni ile 1969’da Noam Chomsky bu yaklaşımı ortaya koydu ve bunun için bazı yapı sistemleri üzerinde çalıştı. Yıllar geçtikçe bu yapı gelişti ve bugünkü şeklini aldı.

3.BÖLÜM: PROGRAMLAMA PARADİGMALARI VE BÖLÜMLERİ

1.İMPERATİVE PROGRAMLAMA

A) PRODECTÜREL PROGRAMLAMA: Genellikle programın işlemleri sırasında kodlama sistemi belirli ve düzenli adımlar silsilesi şeklinde ( Step-by-Step) Yapısı şeklinde ilerler. Bu yaklaşım genellikle ilk kodlamaya başlanıldığı zamanlarda kullanılır.

Avantajları: Özellikle küçük kod blokçularında (Clean code) yapısının uygulanmasını sağlar ve ilk öğrenim sırasında kişilere yardımcı olur.

Yazısal çıktı alabilmemiz için bir fonksiyon girmemiz gerekir.

Dezavantajı: Özellikle büyük sistemlerde güncellenecek bilgi durumu oluştuğunda bunu yapmamız için programda değiştirilmesi gereken yapının çoğalmasıdır.

Örnek diller: C,C+,PASCAL,FARTRON, RUGBY VS.

b)OBJECT ORİENTED PROGLAMLAMA: Gerçek hayattaki nesne ve özellik kavramlarını soyutlama yolu ile işlemler yapılmasını sağlar. Bu sayede nesnenin özellikleri ve fonksiyonlar oluşur.

Avantajları: Nesnelerin özellikler, ihtiyaca göre değiştirilebilir ve programlama için hız, zaman, maddi ölçekte olumlu geri dönütler yapılmasını sağlar.

Vasibility: özellik ABSTRACTİON: SOYUTLAMA

HEREDİTY: KALITIM POLYMARPHİSM: ÇOK BİÇİMLİLİK

Bu sayede sınıfa özel nesne yapabildiğimiz için büyük kod dizilerinde avantaj sağlar

Program çalışmadığı sürece programda yapılacak olan değişiklikler kayıt edilmez. Bu sayede kod blokları arasında ayıklama( debug) yaplıması kolaylaşır.

İstenilen kod bloğu aktif veya pasif hale gelebilir.

OOP hem prosedürel hem de fonksiyonel yapı diilerinin özelliklerini kullanabilir.

DEZAVANTAJLARI: Karmaşık yapısı nedeni ile öğrenme sorunları, bakım zorlukları, işlemci ve ram bellek yapılarına diğer yaklaşımlara göre daha çok kullanması vs.

Örnek Diller: Java, Python, C#, C++, RUGBY VS

C) PARAREL İŞLEME YAKLAŞIMI: Büyük listeli yapılar şeklinde oluşması gereken yapıları bağımsız bir kod dizileri halinde oluşmasını sağlar.

Her bir bileşen belirli bir görevi yerine getirir. Ve bu yapılar minimum etkileşime sahip yapılar oluşur.

Örnek diller: javasciript, Ada, Vusial studio vs

2) DECRALATİVE PROGRAMLAMA

A) FONKSİYONEL PROGRAMLAMA: Bir programın nasıl çalışacağını değil çalışma zamanının durumunu yönetir.

Yapılacak olan kodlamasal işlemleri parça parça ayırarak işlem yapmamızı sağlar.

Bir fonksiyonumuz içerisine değişkenden ziyade direkt içerisine başak bir fonksiyon parametre alabilirse bu fonksiyonel dil ailesine girer.

Avantajı: Dallanma yapısını esas alır.

Fonksiyonların düzenini ve yapısını önemser( geriye değer döndürüdüğü değerlerde vs)

Olumsuz Tarafı: bir fonksiyonun değişimi tüm fonksiyon yapısının değiştirmemize sebep olur.

ÖRNEK DİİLER: HASKELL, FURLANG , HTML, PHP,SQL,LİSP VS

B) MANTIKSAL PROGRAMLAMA: Bilgisayar programlarının mantıksal ifadeler kullanarak açıklama yapan programlama paradigmasıdır. Esas çalışma mantığı önermeler, birinci derece denklemleri kullanarak algoritmasal kurallara uygun çıktı oluşmasını sağlar.

Genel olarak Deklaratif yaklaşım(neyin doğru neyin yanlış olduğunu açıklar)

Kurallar ve gerçekler esas alınır

Sorgulama yapısına esas alınır( program belirli bir sorguya yanıt olarak oluşturulmaya çalışır.

AVANTAJLARI: Yüksek seviyeli çözümleme, kolay bakım olanakları, yapay zeka geliştirmek için esneklik sunması

DEZAVANTAJI: Performans sorunları( büyük projelerde ram ve işlemciye ekstra yük yüklemesi) öğrenme sorunları, birçok uygulama gelişimi için ortam sağlayamaması( oyun, web tasarım vs)

Örnek diller: Prolog, Datalog, asp vs

C) DB YAKLAŞIMI: Açılım olarak database ( veri yapısı) ve veri madenciliği alanlarına yönelik çalışır. Verilerin en verimli şekilde saklanılması, işlenmesi ve geri ihtiyaç olduğunda güvenli şekilde kullanılmasını sağlayan yapılardır.

Veri modelleme, veri depolama, veri erişimi, veri güvenliği veri bütünlüğü, gibi alt dallara ayrılarak hizmet yapar.

Örnek Diller: SQL,PL/SQL, T-SQL, CYPHER, HQL,VS

4.BÖLÜM: Sonuç

Bu yaklaşımlar sayesinde veriler en verimli ( en az ram ve cpu kullanarak en fazla çıktıyı çıkartma çabası) kullanılması amaçlanmış ve temiz kod (Clean Code) yazılımına sahip programalar ihtiyacını gidermeye çalışmıştır. Teknonoji ilerledikçe bu tarz yaklaşımların ilerlemesi ve yeni tarz başak yaklaşımların ortaya çıkması yazılım camiasının ortak kanıya vardıkları nadide durumlardır.

ANLAYIŞINIZ İÇİN TEŞEKKÜR EDER İYİ GÜNLER DİLERİM

ÖĞRENCİ NUMARASI: 0523-051-60-06

ÖĞRENCİ ADI: MELİH IŞIK

ÖDEV HAZIRLANIŞ AMACI: NESNE TABANLI PROGRAMLAMAYA FARLI YAKLAŞIM KAVRAMLARINI ANLAMA VE KAVRAMA

ÖDEV TESLİM TARİHİ 07/10/2024

TESLİM EDİLECEK KİŞİ: DR ÖĞR ÜYESİ SEDA İŞGÜZAR

PROJE SONUÇLARI: NESNE TABANLI PROGRAMLAMA 1.ÖDEVİ