

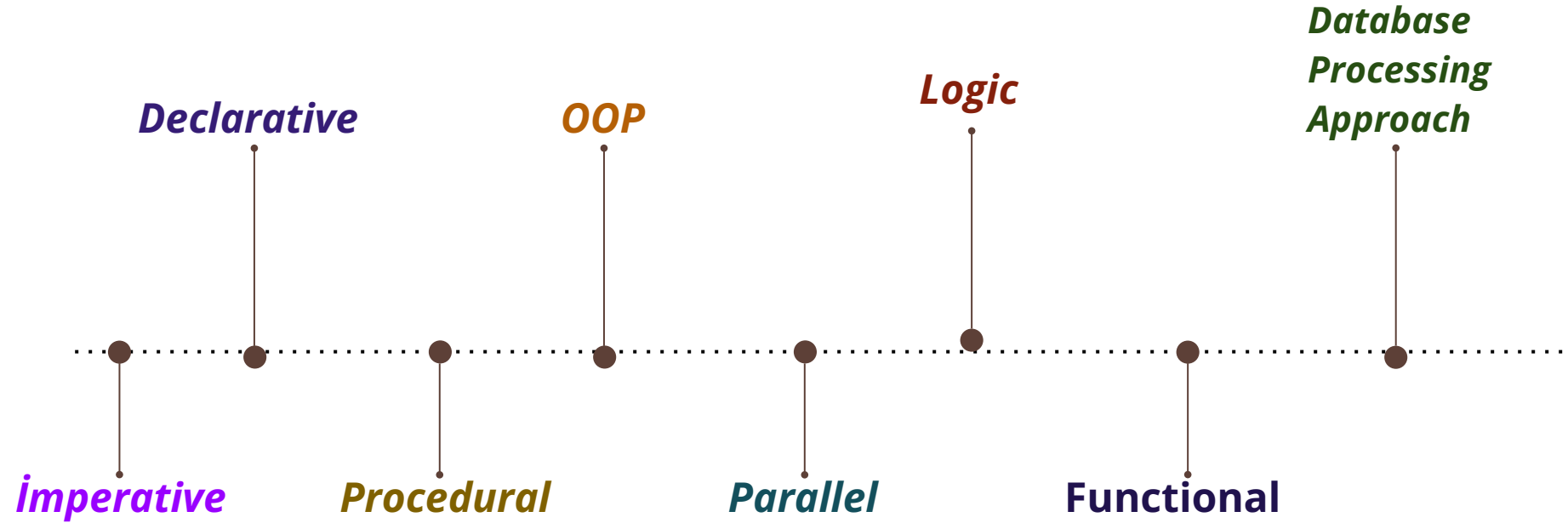


# Paradiqma



*Ramal Fətullayev*

# *Paradigmalar:*



## *İmperative proqramlaşdırma paradiqma:*

İmperative paradiqmada kompüterin yerinə yetirməli olduğu alqoritmləri biz ardıcıl olaraq kompüterə əmrlər verərək icra edirik. Yəni biz komputerə kiçik tapşırıqlar veririk o bu tapşırıqları həll edir və bizə geri hesabat verir.

### **Avantajı:**

- 1.Tətbiq etmək çox sadədir.
- 2.Bu paradigma döngələr, dəyişənlər və s. ibarətdir

### **Dezavantajı:**

- 1.Məhsuldar deyil.
- 2.Paralel proqramlaşdırma mümkün deyil.

**Məsələn:** Dostumuza evimizin yolunu tərif etmək. Burada o bizim ünvana çatmaq üçün hansı yollardan keçməli olduğu barədə məlumat verməyi misal çəkmək olar.



# *Imperative paradigm*



## ***Procedural***

*Proqramların hazırlanması üçün prosedurların addım-addım işlənməsidir. Bu paradigmada kodu təkrar olaraq istifadə edə bilirik.*

## ***OOP***

*Burada sinif və obyekt anlayışları var. Ən kiçik varlıq obyekt hesab olunur və bütün növ hesablamalar obyekt üzərində aparılır. Sinif obyektlərin yaradıldığı plandır. Obyektlər sinifin nümayəndələridir.*

## ***Parallel***

*Paralel emal da işə təlimatlar çoxlu prosessorlar arasında bölünməsi yolu ilə işlənilməsidir. Bu da bir prosesi bölərək daha az müddətə yerinə yetirməyə imkan verir.*

## *Nümunələr:*

### **Procedural**



*Bank hesabı tətbiqi - burada fiziki şəxs üçün hesab, pulun yerləşdirilib çıxarılması üçün hesab, iki fərqli hesab arasında pulun köçürülməsini misal göstərmək olar*

### **OOP**



*Abstraction - Komputerdə oyun oynamaq, film izləmək və s. (İcra hissəsi gizlidir)*  
*Encapsulation - Universitetin daxilində bir neçə şöbənin olmasıdır və bu şöbələr birlikdə universiteti təşkil edir. Polymorphism - Bir adam bir neçə rollarda ola bilər. Məsələn həm tələbə həm də mentor. Inheritance - Uşaqlar valideynlərinin xüsusiyyətini miras alır.*

### **Parallel**



*Bizim eyni anda nəfəs alıb, yazı yazıb, nəyisə izah etməyimiz, kiməsə qulaq asmağımız və s*

## Declarative programlaşdırma paradiqma:

*Bu paradiqma hesablamamanın məntiqini ifadə edən proqramların qurulması üsuludur. Burada proqramçı proqramın necə yetirilməli olduğunu müəyyən etmədən, proqram tərəfindən nəyin yetirilməli olduğunu müəyyənləşdirir.*

*Ümumiyyətlə imperative və declarative paradiqlər fərqi baxsaq, görərik ki, imperative də biz bir şeyi necə edəcəyimizi, declarativedə isə nəyi edəcəyimizi söyləyir.*

### IMPERATIVE

```
// Imperative Programming
let array = [1, 2, 3, 4, 5, 6]
var evenNumbers: [Int] = []

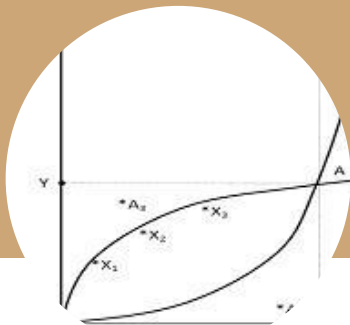
for i in 0..array.count {
  if array[i] % 2 == 0 {
    evenNumbers.append(array[i])
  }
}
```

VS

### DECLARATIVE

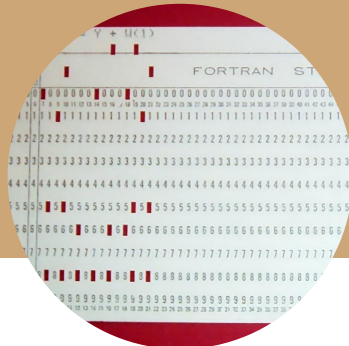
```
// Declarative
let evenNumbers2 = array.filter { $0 % 2 == 0 }
```

# Declarative paradigm



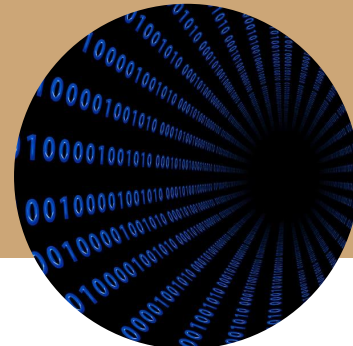
## Functional

*Bu paradigmda əsas riyaziyyatdır və riyazi funksiyaları yerinə yetirir. Bütün kodlar, dəyişənlər funksiya daxilində olur*



## Logic

*Məntiqi proqramlaşdırma paradigması təlimatlardan ibarət deyil, həmçinin faktlar və müddəalardan ibarət olur və bu faktların doğruluğunu yoxlayır*



## Database Processing

*Bu paradigma verilənlər və onların hərəkətinə əsaslanır. Verilənlərin yazılması verilənlər bazasına ötürülməsi lazım olduqda emalı və s.*

## *Nümunələr:*

### **Functional**



*Məsələn bizim gündəlik həyatda pulumuzu sayırıq və nəyə xərclədiyimizi və ya nəyə xərcləsək nə qədər pulumuz qalacağını hesablayırıq .*

### **Logic**



*Məsələn dedektiv oyunlarını misal göstərə bilərəm. Orada faktlar əsasında suçlunu təsbit edib bir nəticəyə varırıq. Amma bu zaman unutmamaq lazımdır ki, bütün faktları nəzərdə keçirdib onun doğruluğunun tutarlı olub olmadığına baxmaq lazımdır.*

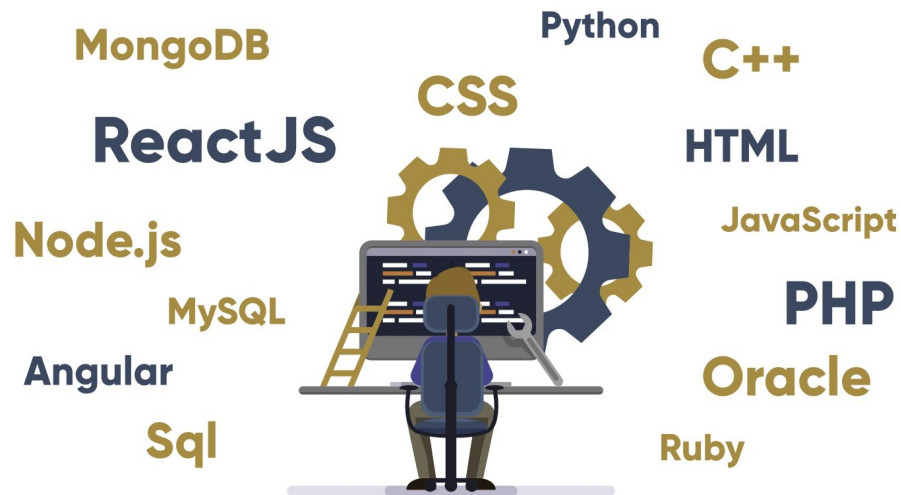
### **Database PA**



*Burda sadə izah olaraq məsələn githubdan qeydiyyatdan keçdikdə məlumatlarımızın və bu tapşırığı ora əlavə etdikdə verilənlərin database köçürülüb saxlanmasını mizsal göstərmək olar.*

---





Diqqətiniz üçün  
Təşəkkürlər

---