1. **Proqramlaşdırma paradigmaları hansılardır.Tam siyahısını yazın**

Programlaşdırma paradigmaları aşağıdakılardır :

* 1. İmperative
  2. Functional
  3. Object Oriented
  4. Parallelism
  5. Logical
  6. Declarative
  7. Procedural
  8. Structured
  9. Event-Driven
  10. Flow Driven
  11. Constraint
  12. Aspect Oriented
  13. Reflective
  14. Array
  15. Encapsulation
  16. Concurrent Computing
  17. Modular programming
  18. Generic
  19. Comparison of multi paradigm
  20. Lazy evaluation
  21. Visual programming language
  22. Concatenative
  23. Prototype-based programming
  24. Function – Level programming
  25. Actor model
  26. Data –driven programming
  27. Stack oriented
  28. Dyanmic
  29. Agent oriented
  30. Role oriented
  31. Interpreted
  32. Dataflow
  33. Strong typing
  34. Markup
  35. Self modifying code
  36. Combinatory logic
  37. Design by contract
  38. Compiled
  39. Multi agent system
  40. Dependent
  41. Type Inference
  42. Knowledge based
  43. Web framwork
  44. Operator

5 paradigma ən əsas sayılır.

1. İmperative
2. Logical
3. Functional
4. Object Oriented
5. Declarative
6. **İmperative ve declarative paradiqmalar arasında 5 ferq yazın**
   1. İmperative paradigmada hər şeyi əllə özümüz edirik. Sanki bir robota əmr verirmiş kimi hər əmri özümüz yazmalıyıq. Burada nəyi necə etməyi də özümüz yaradırıq və göstəririk. Declarative isə nəyi necə edəcəyini özü bilir və özü özünə şərait qurur. Sadəcə nə edilmək lazım olduğunu demək lazımdır. Bunu bir düstur kimidə fikirləşə bilərik. İmperative”də uzun-uzun yazdığımız əmrlər toplusunu, Declarative tək sətirlə həll edə bilər. Məsələn bir insana xəyali bir “Orxan” obrazının evini başa salarkən İmperative paradigmada ona tək-tək “Düz get”, “Sola Dön”, “Filan küçədə filan evin qapısını döy “ tipli bir çox əmrlər toplusu yazmalıyıq. Lakin Declarative paradigmada isə biz “Orxanın evinə get “ demək bəs edir ki, həmin insanı o evə yollayaq. O necə gedəcəyini artıq özü tapır.
   2. İmperative paradigmada özümüz yazdığımız üçün istədiyimiz zaman qurtdalaya, editləyə və əmrləri rahat dəyişə bilərik. Declarative isə öz metodunu özü seçdiyi üçün, bizdə bunu bilmədiyimiz üçün bu dəyişiklikləri edə bilmirik və ya çox çətinlik yaradır
   3. İmperative hər şeyi özümüz etdiyimiz üçün daha çox zamanımızı alacaq. Declarative isə işi qısa zamanda həll edəcək
   4. Declarative programlamada biz İmperative paradigmaya nisbətən daha az xəta edə bilərik.
   5. Declarative paradigma həm də təkrar istifadə etmək üçün də daha yaxşıdır. Eyni kodları təkrarlamamaq, eyni zamanda yazdığımız programı sürətləndirə bilərik
7. **Obyekt yönümlü paradiqma haqqında real nümuneler göstərin (3 ədəd)**
   1. Ətirlər. Hamsı bir obyektdir və bir məqsəd üçün lazımdır. Lakin bir-birindən öz xüsussiyyətlərinə görə fərqlənir. İy fərqi, qramı, rəngi, az iyli və kəskin iyli olması və .s onları bir birindən ayıran xüsusiyyətlər vardır
   2. Maşınlar. Maşın dediyimiz obyektlərin hamsı bir məqsəd üçündür ancaq onları da müxtəlif markalara, at güclərinə, rənglərinə, müxtəlif xüsusiyyətlərinə görə ayırırıq.
   3. Telefonlar
8. **Proqramlaşdırma paradiqmalarında hansını özünüzə daha yaxın hesab edirsiniz? Səbəbini detallı şəkildə izah edin..**

Mən daha çox İmperative paradigmanı seçərdim. Çünki bütün əmrləri özümüz veririk və bütün idarə özümüzdədir. Digər paradigmalara nisbətən daha rahat dəyişdirilə bilər və daha azaddır. Əmrlərin necə edildiyini də bildiyimiz üçün hər şeyi idarə etmək daha rahatdır.

<https://e-bergi.com/y/programlama-paradigmalari/>

<https://www.quora.com/What-is-the-meaning-of-a-programming-paradigm>

<http://cs.lmu.edu/~ray/notes/paradigms/>

<https://www.info.ucl.ac.be/~pvr/VanRoyChapter.pdf>

<https://www.geeksforgeeks.org/introduction-of-programming-paradigms/>

<https://medium.com/@darrion/what-is-meant-by-programming-paradigms-9b965a62b7c7>

<https://tylermcginnis.com/imperative-vs-declarative-programming/>

<https://blog.webix.com/difference-between-declarative-and-imperative-programming-with-language-examples/>

<https://hub.packtpub.com/what-difference-between-declarative-and-imperative-programming/>