**Paradigma**

* programlasdirmaya yanasma novudur
* Structure qurma novudur.
* Riyazi teoremlere ve iliskili prinsiplere esaslanir.

Program language (dili) - problemin cozumu üçün executable (çalışa bilən) modellərin qurulması üçün tool (vasitedir) dir.

**Paradigma novleri** :

* Imperativ
* Funksional
* Logical
* Object oriented (OO)

**Imperative Paradigma (ƏMR)**

Əməliyyatlar belli ardicil əmrlər funksiyalar və ya ardıcıl etaplar şəklində həyata keçir. Siralama cox əsasdır. Dəyişənlər mühitində baş vermiş dəyişiklikləri işləm prosesinde qoruyub saxlayir

Musbet ceheti – masin dili isleme prinsipine benzerliyi vardir.

İmperative Paradigmanin bir növüdə Structural programming di. Axiş burada “iç-içə” (nested) loop(döngü) və conditions(şərtlərlə) idarə edilir nəinki GoTo la

**Functional Paradigm**

Butun subprogramlara funksiya kimi baxılır – argumenti götürüb dəyər dönürlər. Dönülən (return) dəyər daxil olan (argument) asılıdır. Eyni funksiyani eyni deyerle bir nece defe cagirsalar belə eyni dəyəri dönməkdədir. Yan təsirləri neytrallaşdirir (eliminate side effects)

Xususiyyetleri

* Asılılığı yoxdur
* Assignmentler yoxdur
* Funksiya gondererek (argument olaraq) funksiya əldə etmək olar
* Komandalar yoxdur
* Kod qisadir

**Logic Paradigma**

Formal logic-e esaslanmişdir. Programlar fakt ve qaydaların mentigi giymetlendirme uzerinde qurulur.Butun faktlar müeyyenlendirilir. Komputerin işi datanı saxlamaq, ve deduksiyani heyata keçirmektir.

Object -oriented paradigm (OO)

Bu paradigmada real objectler ayri vahidler kimi dusunulur. Object ozunde data ve metodlari ehtiva eden yazilim vahidi ve ya composite data typedir.

Objectlerle komnukasiya onlara deyer gondermekle elaqe qurulur.

Class - objectin abstract tesevvurudu ve ya objectin strukturudur.

Class ( data type) - composite data type like cat

{ - field (data member) - state – colour,name (adi)

-method (function member) } – behavior (davranis) ----gezmek , yemek ve s.

Principles

* Encapsulation
* Inheritance
* Abstraction

**Abstraction -**

Abstraction dedikde obyekti bezi xususiyyetleri olan ve islemler eden bir data type kimi umumilesdirmekdir.

Her bir objection xususiyyetleri vardir. meselen Binadirsa hundurluyu fasadi ve s. Bu xususiyyetler binani temsil edir. Objectin heyata kecirdiyi islere funksiyalarasa metod deyilir.

Abstraction Class bazli edilir. Class icinde o objecte ozellikler ve metodlar tanimlanir.

Class abstract struct oldugundan bir basa istifade edile bilmez, bu sebebden o sinifden ornekler numayendeler qurulur (instance)

**Inheritance principi**

OO progralamada bir object umumilikde bir object sinifina aid bir numunedir.-( Instance)

Animal classindan cat objecti yaradilarsa (instance) o animal classinin sahib oldugu ozelliklere sahibdir ve onun yerine yetire bildiyi funksiyalari yerine yetirir.

Ozunden alt sinif qura bilen sinifa base class, parent class vey a parent class deyilir.

Subclass ise child class deyilir.

Derived class elave olaraq basqa stateleride ola biler. Yeni cat objectinin animal classinda oldugu ozelliklerle yanasi meselen tullanmaq funkisyasida yerine yetire biler.

Inheritance ustunluyu cod istifadenin onune gecir.

Encapsulation

Encapsulation – class I mueyyen eden metod ve xususiyyetlerin (class =data+metod) classi istifade eden kullanicilardan gizlemesidir.

Public – xarice aciq , hec bir informasiya gizledilmir.

Private – xaricden hec bir informasiyaya baglanti yoxdu

Protected - yalniz child classlar bu informasiyaya erise bilir

Polymorphism

Ferqli type input code eyni olarsa xeta verir. Output vermez

Amma polymorphism zamani eyni cod olsada fergli inputlar ucun fergli outputlar cixir.