### DESIGN PRINCIPLES AND PATTERNS:

#### DESIGN PRINCIPLES

5 - Single xesponsibility Pxinciple.

O - oPen/ closed Principle

Company Le Liskov Substitution Principle

I - Interface Segregation Principle.

D - Dependency Inversion Principle

# Introduction to Solid Principles:-

- -> Solid Principles one introduced by Robert c. Martin in his 2000 Paper "Design Patterns and Design Principles"
- -7 Latex built by Michel Feathers who introduced sollD ackonym.
- These 5 Principles Sevolutionized the object oriented Prog-Samming changing the way we warte software
- -> Maxtin of feathers " Design Principles of design Patterns" encouxage us to execte more maintainable, understandable and flexible soft wore
  - As our application grows these principles helps us to se duce complexity.

#### Single Responsibility Principle:-

-> SRP (Single xesponsibility Pxinciple) state that "A class Should only have one responsibility. Furthermore it should have only one xeason to change",

How does SRP helps us to buil better softwere:

[1] Testing: A class with one responsibility have fewer TC. [2] Lower coupling: A class with one xesponsibility have fewer dependen

[3] Oxanization. Smaller, well organised casses are easily to search.

```
Example: - (with SPP Principal)!
Class
                             11 Animol methods.
     Zoo Entity
{
                                 Void Sleeped;
   11 Staff
                                 Void
                                      cat (),
         vawaine faller dualist star
    String
                      March Mar Void Aly C);
          acutes symme
    Stking
                             vancia void fight ();
          036
    int
    Double salony.
                                  11 Visitos methods.
    String de Partment:
                                  void catesi
                                  Void Sound & Sound c )
    1 Animal
    sexing name:
                                        click Pictures co;
    String gender;
   ante nagerania da exemens polos vo dina como
    String species.
                              Problems with above code:
    boolean canFIY;
                  on booms of > Variable naming Confrit
    boolean cats Meat; -7 Difficult to test.
    11 visitors. Tightly couried in
    String name; John 9 -7 Comes under Violation of SRP.
     int age;
                            and Flexible Scheucoid
     string gender.
                  our springlion arous those P
    Long Ticket Id,
                               Tion To hirs Complexity
     Date Time Date;
Methods
                   28 SER ( Tagle Restranbilley Principle)
     11 staff methods.
           Sleep cy; Villde mayor one orad for Books
     Void
           ester; "ormers or most one for shad
      Void
      Void
           walk()
           feed Animal ();
     void
           clean Premises C).
     Void
```

```
code :- (with SRP Principio).
class 700 Entity class staff extends 700 Entity
 string name;
                             doubic
                                    salary;
                                    designation,
  String genler,
                           String
  int age;
                             Void feed Animal (2):
Void catco;
                             Void clean Premises co;
  Voida sice PCD, and to 7 soldo dans
          moxeste of the covertues of the biscoria
class Animal extends too Entity class Vistor Extends with the
                                         Williams Statement A. A.
                                       Strim ticket Idc).
to bookinsticonFIX VIANDAND So 42007
                                       Date Time date;
      boolan eatsmeat
                                       void Koam Akoun des,
      void fight();
                             lomin void > click Pictorescy-
      void fly CD;
         Closed Pxinciple:-
                                     Lateston Iconfly ()
 -7 open closed principle states that "A class should open
 for extension but closed for modification.
                                  impost Zoolibsony. Bixd
 [Public 700 Libxary]
                 CASS Trackous extens plus implorents
                                   Class Awesome
    class Animal 2
                                       void main c) &
      String Specifies.
    7
                               boild Bisd Bit new Bisd ("Piggon")
    class Bird Extends Animal [
                                   3 B. fy();
     roid flyer {
         if (species =="spakow") . ..
      else if (species= Pegion) ...
                                 -> what if client add new Bixe
                                 -7 Extendion is not ressible
      else if (species = = eagle")..
                                    if we use if-else
                                     laddex.
```

```
[3] Liskov substitution Principle:
 -7 An object of Parent class should be seplicable with
  any object of child class extends pasent without
  Causing exxox.
                       (ox)
-> If s is a subtype of T, the object of type T
   may be seplaced with object of type s without
   altering the correctness of the Program.
7 A subclass should behave in a way that it should not
   break the expectations set by its Parent class
-7 The desired class most be completly substituable for
       base class.
                                 Jours is shell in
 Class Bird extends Animal
    abstract void cot ();
  Interface IcanFIY ()
  Though sound eximite stated that " y this should
       void fixc) to to he of book to porm to book
  3
       sportsow extent Bird implements
                                      I confly of
                                    Maria Aspend
                          Station over 112 de
       TO MIRK FIRE
  . Class Kiwi extends Bixd { prim change by
                                  I CYIT TIME
                           . . (Charter - Esport) Y
                             (mane, come) gi selo
```

· ( 011 100 , - 57 1 005 ) 41 0013.

## Intexface Segregation:

ITSP States that longe interfaces should be split into Smaller one8.

of By doing this we can ensure that implementing classes only need to concerned about the methods.

Public intexface Bearkeeper

{

Void Feed The Bearc?

Yold wash The Bearc?

Public intexface Bearfeeder

{

Void FeedTheBearcz.}

}

Public intexface Bear Cleaner {

Void washTheBearcz.}

#### [5] Dependency Inversion:

- -> Dependency Inversion Principle refers to decoupling of software modules.
  - -7 High level modules should not depend on low-level modules.
- -7 Abstraction should not depend on details.
  - -> Details should depend on abstraction.