SOLID

Wednesday, 26 May 2021 9:05 PM

- S : Single responsibility Prin
- O : Open - close prin
- L : Liscov Substitution
- I : Interface segregati
- D : Dependency Inversion

SRP

Class Bird {

void fly (type) {

= if (type == "hen") {

//

}

= else if (type == "eagle") {

// ———

}

;

"else if (type == "flamyo") {

//

50

void eat ( ) {

50

}

void layEgg ( ) {

50

}

Bird

Hen

Eagle

Flamyo

fly ( ) ;

eat ( ) ;

layEgg ( ) ;

Lutmus test

pH

0 ++ 14

6 7 8

① Multiple if / else / Switch-case

→ Behavior

② Monster Method

int getIncome ( ) {

i = e.incom ;

// generate Pay slips

// Ps → Json

// Mail PS

retn i ;

}

③ Unspecified Util / helper class.

class Util {

int rupee To Dola (a) ;

int rndOff Double (d) ;

int gethash (obj) ;

int calculateIT (e) ;

Strig to JSONSt (obj) ;

}

OCP

Open for extention

Closed for modification

× { Refacting, Bug fixes, Updation }

LSP

The objects of parent class should be completely replaceable by objects of child classes.

Bird

fly ( )

eat ( )

layEgg ( )

→ Interface

Hen

Kiwi

Eagle

fly ( ) {

// Exception (. .)

interface flyable { LSP

void fly ( ) ;

void flapWing ( ) ;

Hen

Eagle

Money

Kuda

Mig-21

Interface Seg. Prin

(SRP # LSP)