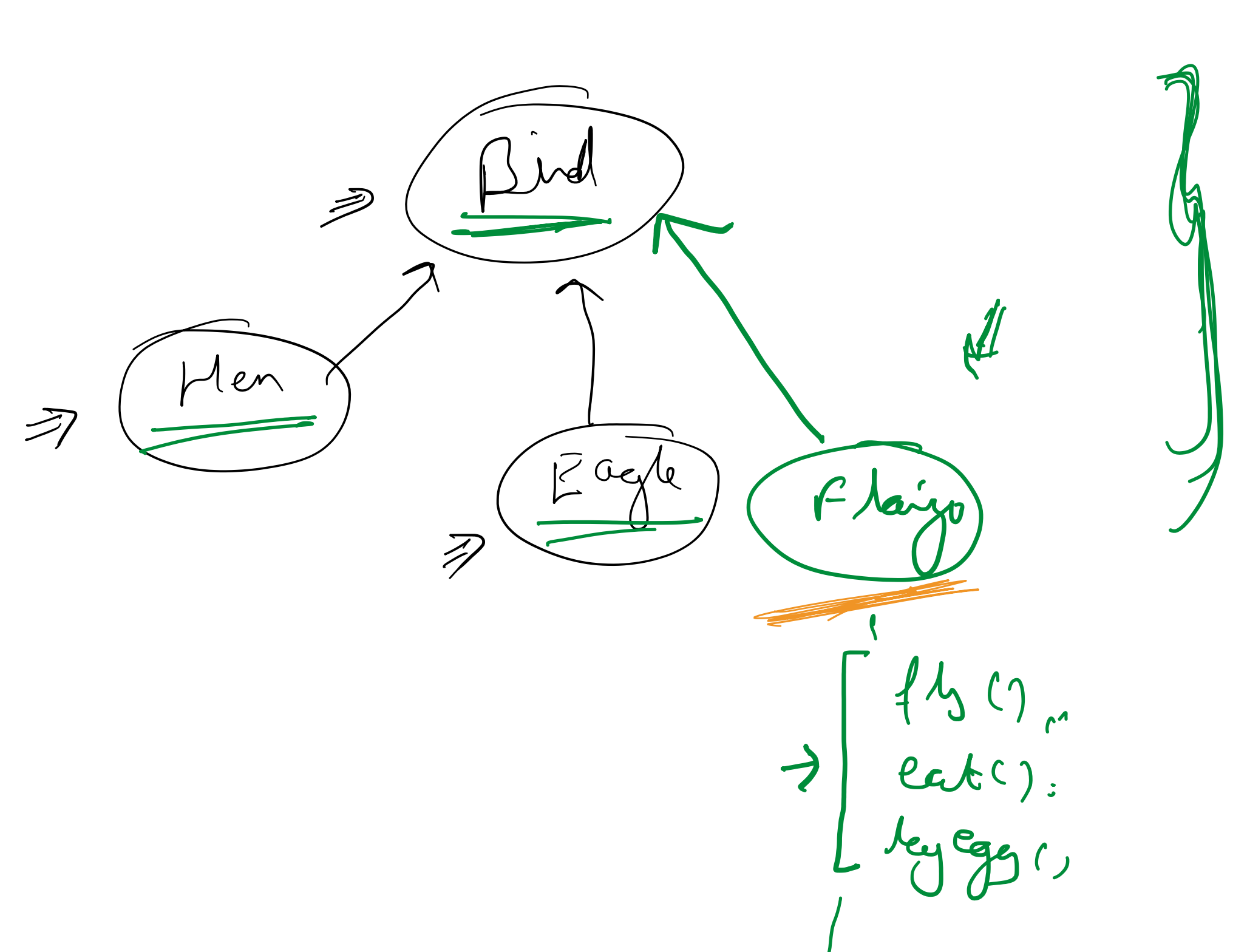


- S : Single responsibility Pr
- O : Open-close pr
- L : Liskov Substitution
- I : Interface segregation
- D : Dependency Inversion

SRP

```
class Bird {  
    void fly (type) {  
        //  
        if (type == "hen") {  
            //  
        }  
        else if (type == "eagle") {  
            //  
        }  
        else if (type == "flango") {  
            //  
        }  
    }  
    void eat () {  
        //  
    }  
    void layEgg () {  
        //  
    }  
}
```

↳ PH 0 1 2 3 4 5 6 7 8 9 10



① Multiple if/else/Switch-case
↳ Behavior

② Monster Method

```
int getIncome () {  
    i = e.income;  
    // generate Pay slips  
    // ps -> JSON  
    // Mail ps  
    return i;  
}
```

③ Unspecified Util/helper class

```
class Util {  
    int rupeeToDollar (a) {  
        //  
    }  
    int roundOffDouble (d) {  
        //  
    }  
    int gethash (obj) {  
        //  
    }  
    int calculateIT (e) {  
        //  
    }  
    String toJSONStr (obj) {  
        //  
    }  
}
```

OCP

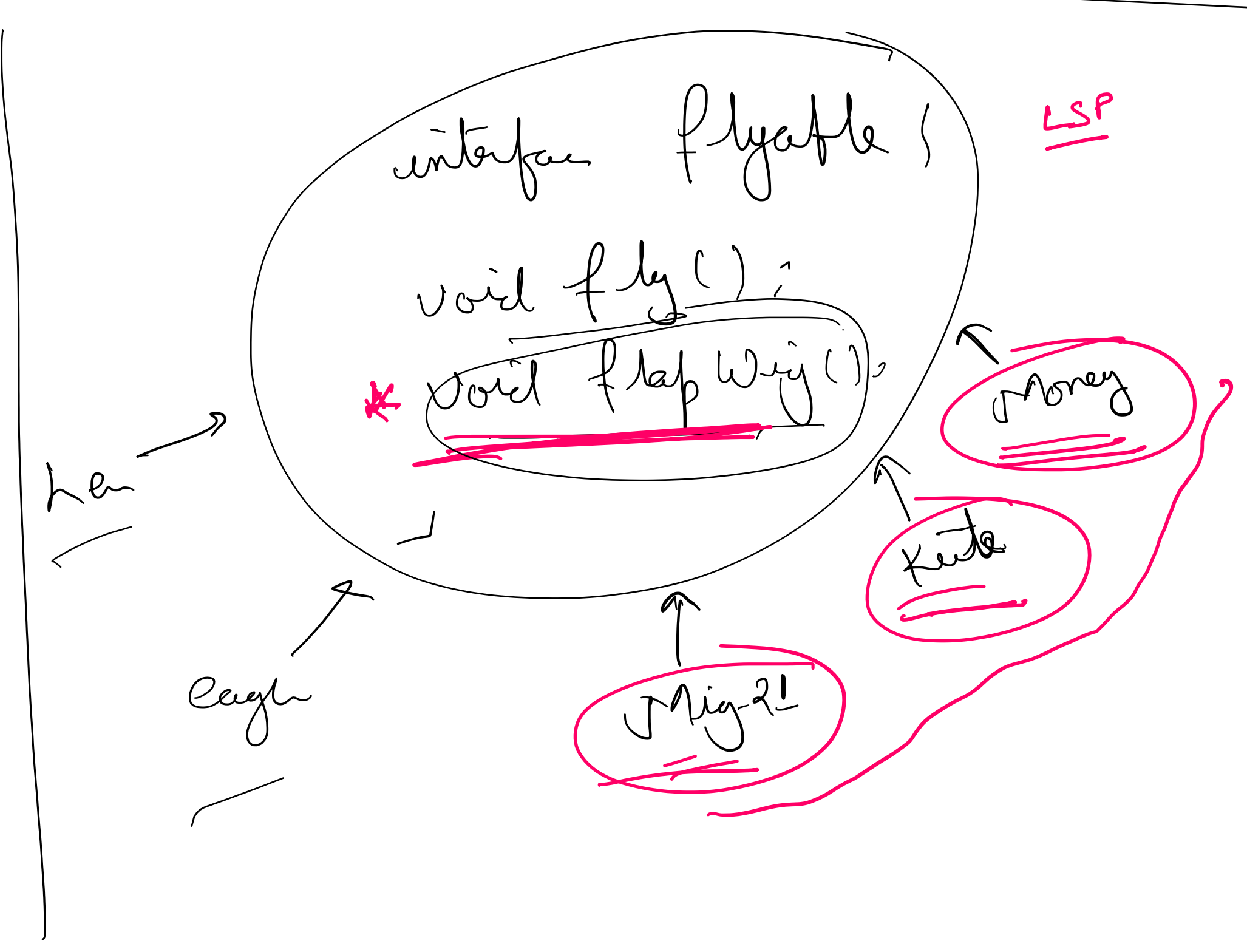
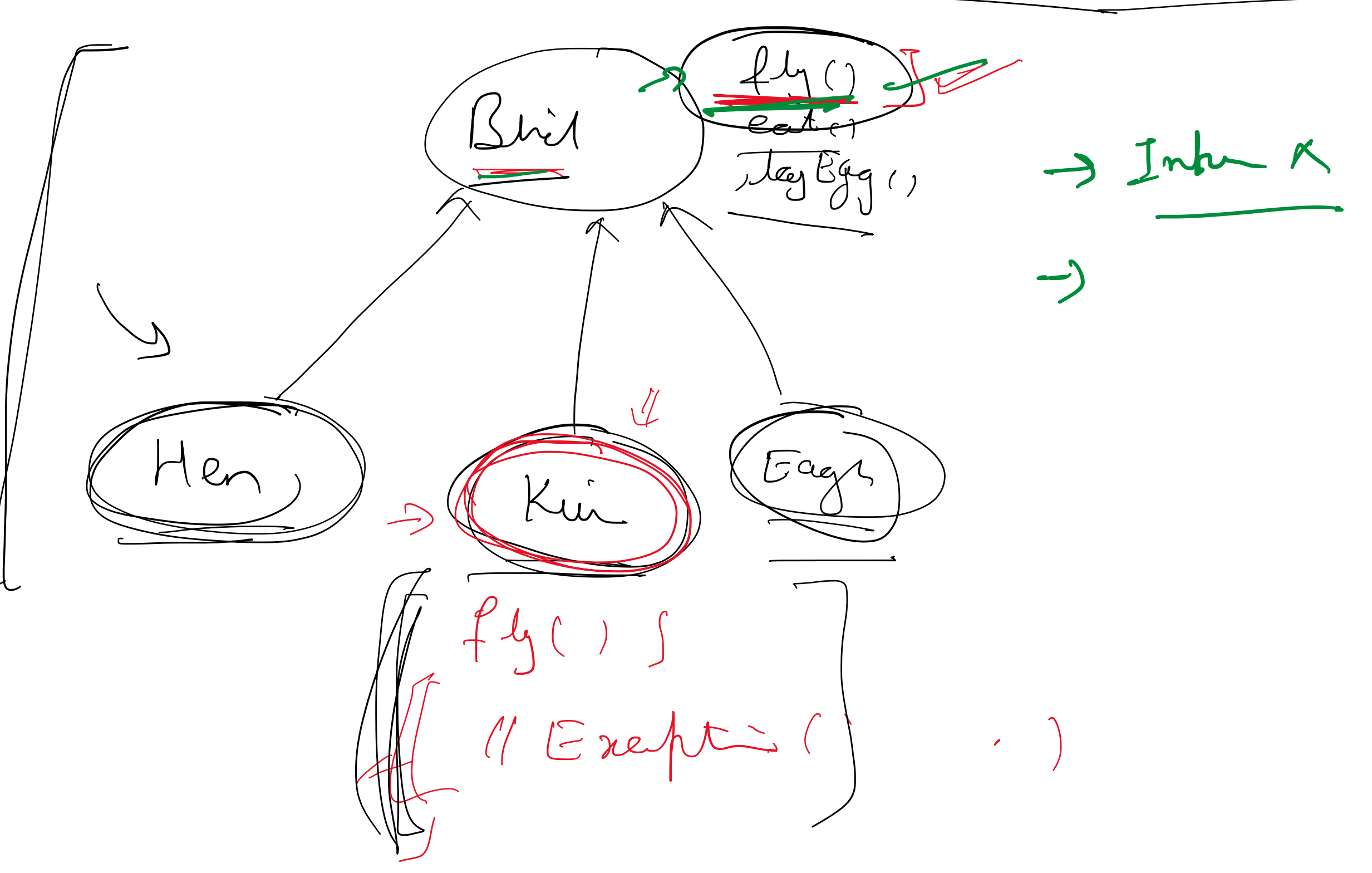
Open for extension
Closed for modification

Refactoring

Bug fixes

Updates

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



Interface Seg. Pr

(SRP & LSP)