

Day - 05 Solid Principles

SOLID 1: SRP and OCP

- ① SOLID
- ② Single Responsibility (SRP)
- ③ open-closed (OCP)
- ④ Real life design use case.

Good Software

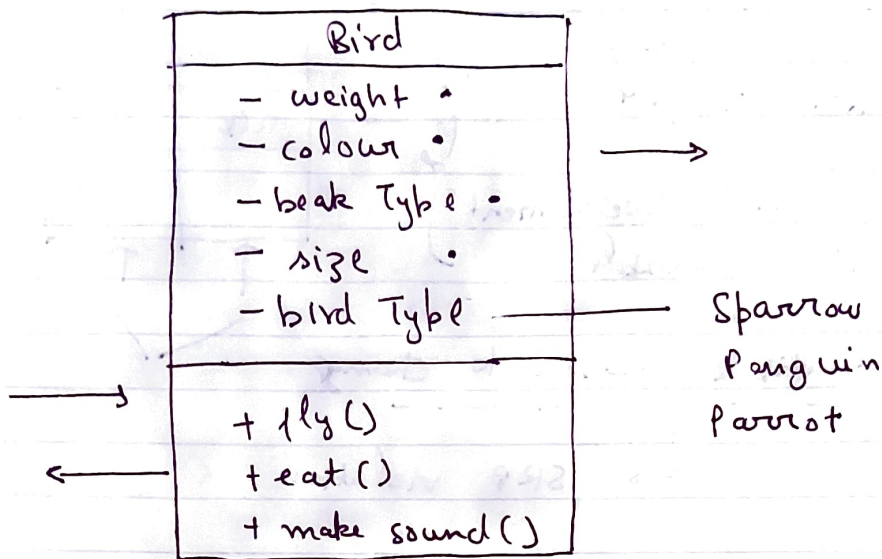
↳ maintainable. } → SOLID.
↳ extensible.
↳ scalable

Single responsibility (SRP)

↳ class should do one thing.

↳ one reason to change.

-
- ① Real world — Design a bird.
 - ② Games.
 - ③ Management S.



```

class Bird {

```

```

    public void fly() {

```

```

        if (bird is parrot)

```

```

            fly like a parrot

```

```

        else if (bird is sparrow) —

```

```

            — — — — —

```

```

        else if (Penguin)

```

```

            → Dont fly

```

```

    }

```

```

}

```

```

= if-else hell =

```

- ① Not readable
 - ② Not testable
 - ③ Parallel development
 - ④ Not reusable
 - ⑤ Multiple reason to change
- ↳ SRP violation.



SRP benefits

- ① Maintainable — easy debug.
- ② Lower coupling
- ③ Easy to test

How to spot SRP violation!

- ① If one code has multiple reason to change.
- ② If - else hell.
- ③ God - Class } Lot of code.
Monster method }

```

public void saveToDatabase {
    // connect to a database.
    // create a query
    // execute that query.
    // create a user obj
    // close the collection
}
  
```

④ utility classes

open closed

- ↳ class should be
 - open for extension
 - closed for modification.

```
fly() {
```

```
  if (type == sparrow) {
```

```
    ...
```

```
  } else if (Penguin) {
```

```
    ...
```

```
  } else if ( ) {
```

```
}
```

Bird → Penguin

Parrot Sparrow.

