

Interface Segregation Principle

Peter Öttl

SOLID

- S - Single Responsibility Principle
- O - Open Close Principle
- L - Leskov Substitution Principle
- I - Interface Segregation Principle
- D - Dependency Inversion Principle



Interface Segregation Principle

When more means less

Interface Segregation Principle

- große Schnittstellen in mehrere Schnittstellen aufgeteilt
- schlanken Schnittstellen
- verbesserte Wartbarkeit und Testbarkeit
- Code ist kompakt und besser wiederverwertbar
- bessere Überprüfbarkeit
- Keine unnötig implementierten Methoden

```

interface Bird {
    fly(): void;
    walk(): void;
}

class Duck extends Bird{
    fly(){
        // Duck can fly
    }
    walk(){
        // Duck can walk
    }
}

class Ostrich extends Bird{
    fly(){
        // Ostrich cant fly... throw some error
    }
    walk(){
        // Ostrich can walk
    }
}

```

```

interface BirdFly{
    fly(): void;
}

interface BirdWalk{
    walk(): void;
}

class Duck extends BirdFly, BirdWalk{
    fly(){
        // Duck can fly
    }
    walk(){
        // Duck can walk
    }
}

class Ostrich extends BirdWalk{
    walk(){
        // Ostrich can walk
    }
}

```