INTERFACE SEGREGATION

```php
interface IItem
{
    public function applyDiscount($discount);
    public function applyPromocode($promocode);

    public function setColor($color);
    public function setSize($size);

    public function setCondition($condition);
    public function setPrice($price);
}
```

```
interface IItem
{
    public function setCondition($condition);
    public function setPrice($price);
}
```

```
interface IClothes
{
    public function setColor($color);
    public function setSize($size);
    public function setMaterial($material);
}
```

```
interface IDiscountable
{
    public function applyDiscount($discount);
    public function applyPromocode($promocode);
}
```

```
class Book implemets IItem, IDiscountable
{
    public function setCondition($condition){/
*...*/}
    public function setPrice($price){/*...*/}
    public function applyDiscount($discount){/
*...*/}
    public function applyPromocode($promocode)
{/*...*/}
}
```

```
class KidsClothes implemets IItem, IClothes
{
    public function
setCondition($condition){/*...*/}
    public function setPrice($price){/
*...*/}
    public function setColor($color){/
*...*/}
    public function setSize($size){/*...*/}
    public function setMaterial($material)
{/*...*/}
}
```

INTERFACE SEGREGATION

«Много специализированных интерфейсов лучше, чем один универсальный». Соблюдение этого принципа необходимо для того, чтобы классы-клиенты использующий/реализующий интерфейс знали только о тех методах, которые они используют, что ведёт к уменьшению количества неиспользуемого кода.

```
interface IDiscountable
{
    public function applyDiscount($discount);
    public function applyPromocode($promocode);
}

class Book implemets IItem, IDiscountable
{
    public function setCondition($condition){/
*...*/}
    public function setPrice($price){/*...*/}
    public function applyDiscount($discount){/
*...*/}
    public function applyPromocode($promocode)
{/*...*/}
}
```

```
class KidsClothes implemets IItem, IClothes
{
    public function
setCondition($condition){/*...*/}
    public function setPrice($price){/
*...*/}
    public function setColor($color){/
*...*/}
    public function setSize($size){/*...*/}
    public function setMaterial($material)
{/*...*/}
}
```

DEPENDENCY INVERTION

```
}
{\*...*\}
    public function applyPromocode($promocode)
*...*\}
    public function applyDiscount($discount){\
    public function setPrice($price){\*...*\}
*...*\}
    public function setCondition($condition){\
{
class Book implements ItemInterface
```

```
}
{\*...*\}
    public function setMaterial($material)
    public function setSize($size){\*...*\}
*...*\}
    public function setColor($color){\
*...*\}
    public function setPrice($price){\
```