

LLSKOV/STUBSTUTION




```
function calculateRectangleSquare(Rectangle $rectangle, $width, $height)
{
    $rectangle->setWidth($width);
    $rectangle->setHeight($height);
    return $rectangle->getHeight * $rectangle->getWidth;
}
```

```
calculateRectangleSquare(new Rectangle, 4, 5); // 20
```

```
calculateRectangleSquare(new Square, 4, 5); // 25 ???
```

LISKOV SUBSTITUTION

Принцип гласит – «Объекты в программе могут быть заменены их наследниками без изменения свойств программы». Своими словами я бы это сказал так – при использовании наследника класса результат выполнения кода должен быть предсказуем и не изменять свойств метод. К сожалению, придумать доступного примера для этого принципа в рамках задачи интернет-магазина я не смог, но есть классический пример с иерархией геометрических фигур и вычисления площади. Код примера ниже.

```
function calculateRectangleSquare(Rectangle $rectangle, $width, $height)
{
    $rectangle->setWidth($width);
    $rectangle->setHeight($height);
    return $rectangle->getHeight * $rectangle->getWidth;
}
```

```
calculateRectangleSquare(new Rectangle, 4, 5); // 20
calculateRectangleSquare(new Square, 4, 5); // 25 ???
```

```
class Rectangle
{
    protected $width;
    protected $height;

    public setWidth($width)
    {
        $this->width = $width;
    }

    public setHeight($height)
    {
        $this->height = $height;
    }

    public function getWidth()
    {
        return $this->width;
    }

    public function getHeight()
    {
        return $this->height;
    }
}
```

```
class Square extends Rectangle
{
    public setWidth($width)
    {
        parent::setWidth($width);
        parent::setHeight($width);
    }

    public setHeight($height)
    {
        parent::setHeight($height);
        parent::setWidth($height);
    }
}
```


