

# Принцип единственной ответственности

**№ урока:** 1    **Курс:** SOLID принципы в Java

**Средства обучения:** IntelliJ IDEA, Eclipse

## Обзор, цель и назначение урока

Цель урока – ознакомление с понятием SOLID. Подробное рассмотрение принципа единственной ответственности.

## Изучив материал данного занятия, учащийся сможет:

- Использовать один из принципов SOLID.
- Понимать архитектуру приложения на языке программирования Java.
- Проектировать приложение согласно принципам SOLID.
- Отличать “плохой код” от хорошего кода согласно одному из принципов

## Содержание урока

1. Признаки плохого кода.
2. История создания акронима.
3. Введение в SOLID.
4. Пользовательские директивы.
5. Архитектура проекта
6. Принцип единственной ответственности.
7. Рассмотрение примеров

## Резюме

На данном уроке студенты познакомились с понятием SOLID. Рассмотрели отличие плохого кода и хорошего, а также узнали какие бывают этапы разработки программного обеспечения. Учащиеся узнали понятие принципа единственной ответственности и его использования. Также познакомились с написанием кода согласно этому принципу.

Принципы SOLID — это стандарт программирования, который все разработчики должны хорошо понимать, чтобы избежать создания плохой архитектуры. Этот стандарт широко используется в ООП. Если применять его правильно, он делает код более расширяемым, логичным и читабельным. Когда разработчик создаёт приложение, руководствуясь плохой архитектурой, код получается негибким, даже небольшие изменения в нём могут привести к некорректной работе приложения. Поэтому нужно следовать принципам SOLID.

**Принцип единственной ответственности** — *“A module should be responsible to one, and only one, actor.”* («Модуль должен быть ответственным за действия, которые будут выполняться одной и только одной ролью»).

Эти проблемы возникают, потому что мы ставим код, который предназначен для разных ролей в непосредственной близости. SRP говорит отделить код, от которого зависят роли.

## Закрепление материала

- Кто впервые описал эти принципы?
- Какие существуют признаки плохого кода?
- Кто придумал акроним?
- О чем гласит принцип единственной ответственности?

- Чем отличается старая формулировка принципа от новой?
- Какие преимущества использования данного принципа?

### Самостоятельная деятельность учащегося

#### Задание 1

Выучить основные понятия, рассмотренные на уроке.

#### Задание 2

Попытайтесь разделить код, который находится в пакете homework\_task таким образом, чтобы код соответствовал принципу единственной ответственности.

### Рекомендуемые ресурсы

["Clean Architecture" Robert C. Martin](#)

<https://vimeo.com/43592685>