1. **SOLID** - это принципы разработки программного обеспечения, следуя которым Вы получите хороший код, который в дальнейшем будет хорошо масштабироваться и поддерживаться в рабочем состоянии.

**S - Single Responsibility Principle - принцип единственной ответственности**. Каждый класс должен иметь только одну зону ответственности.

**O - Open closed Principle - принцип открытости-закрытости.** Классы должны быть открыты для расширения, но закрыты для изменения.

**L - Liskov substitution Principle - принцип подстановки Барбары Лисков.** Должна быть возможность вместо базового (родительского) типа (класса) подставить любой его подтип (класс-наследник), при этом работа программы не должна измениться.

**I -  Interface Segregation Principle - принцип разделения интерфейсов.** Данный принцип обозначает, что не нужно заставлять клиента (класс) реализовывать интерфейс, который не имеет к нему отношения.

**D - Dependency Inversion Principle - принцип инверсии зависимостей.** Модули верхнего уровня не должны зависеть от модулей нижнего уровня. И те, и другие должны зависеть от абстракции. Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.

1. Хотя мы можем создать обычный класс, который не является наследником, но фактически все классы наследуются от класса **Object**. Все остальные классы, даже те, которые мы добавляем в свой проект, являются неявно производными от класса Object. Поэтому все типы и классы могут реализовать те методы, которые определены в классе Object. Рассмотрим эти методы.

toString(), hashCode(), equals()

1. Простое наследование — это механизм, при котором класс (наследник) наследует свойства и методы только от одного родительского класса. В Java реализовано простое наследование, где каждый класс может иметь только одного родителя.

Множественное наследование — это возможность класса наследовать свойства и методы от нескольких родительских классов. В Java множественное наследование не поддерживается напрямую для классов, чтобы избежать сложностей, связанных с конфликтами, например, "алмаза". Однако Java поддерживает множественное наследование через интерфейсы.

▎Особенности реализации наследования в Java

1. Класс Object: Все классы в Java наследуются от класса Object.

2. Ключевое слово extends: Используется для указания родительского класса.

3. Ключевое слово super: Используется для обращения к родительскому классу.

4. Переопределение методов: Подкласс может переопределить методы родительского класса.