**Принципы ООП**

# **Инкапсуляция**

* Сокрытие деталей реализации и поведения объекта от его клиентов

На состояние робота мы можем влиять только через специальные методы. Мы вызываем метод, который отвечает за какое-то действие робота, а он в свою очередь использует другие приватные методы, сокрытые от клиента.

* Проекция правил взаимодействия с объектами в реальном мире на проектируемый объект

# В реальном мире, например, мы не можем телепортировать робота в другое место, чтоб изменились его координаты, мы должны выполнить движение.

# **Наследование**

* Устранение дублирования спецификации классов
* Отражение типа связи “является” (иерархии)
* Наследование интерфейса

# Есть разные роботы, которые фотографируют, и снимают видео, но и те и другие имеют общие функции, например движение, повороты, местоположение в пространстве. Но есть также роботы, которые могут и фотографировать и видео снимать, используя интерфейсы можно продемонстрировать множественное наследование. Таким образом устраняется дублирование спецификации классов.

# **Полиморфизм**

* Разное поведение объектов поддерживающих общий интерфейс в зависимости от их реальных типов

Можно задать движение роботу, указав расстояние, на которое он должен переместиться, можно указать кроме расстояния еще и направление, или просто указать направление. Один интерфейс - разные поведения.

* Ключевым в использовании полиморфизма является работа клиентского кода с интерфейсом не имея информации о реально использованных классах реализующих этот интерфейс

Есть интерфейс Команда, его реализуют классы разных команд. Для того чтоб получить любую конкретную команду мы работаем с одним и тем же интерфейсом Команда.