

Тестирование

- *Автоматизированное тестирование* предполагает наличие программ, работающих через функции с возвращаемыми значениями
- Если в нашей программе нужно обратиться к базе данных и другим сервисам, которые еще не написаны – используются *Mock-объекты*
- Нельзя создавать объекты классов внутри методов других классов → принципы *IOC/DI* → начинаем с программирования интерфейса и в методы передаем классы → если захочется поменять например “место логирования”, нам нужно будет поменять метод этого объекта логера
- unittest – стандартный фреймворк для тестирования
 - вызываем `unittest.main()` → вызывается объект наследуемый от `unittest.TestCase`
 - в методах класса вызываются `assert*` функции
 - `setUp/tearDown` – создание общего объекта/освобождение ресурсов
 - Объединять тесты
 - Пропускать тесты
- TDD – test-driven-development – сначала пишется тестирование → Потом программа, покрывающая эти тесты и похожие к ним
- BDD – behaviour-driven-development → Тесты будет писать не программирующий пользователь, а заказчик на псевдоестественном языке.
- Шаблон проектирования – повторяемая архитектурная конструкция: [Сайт паттернов](#)