

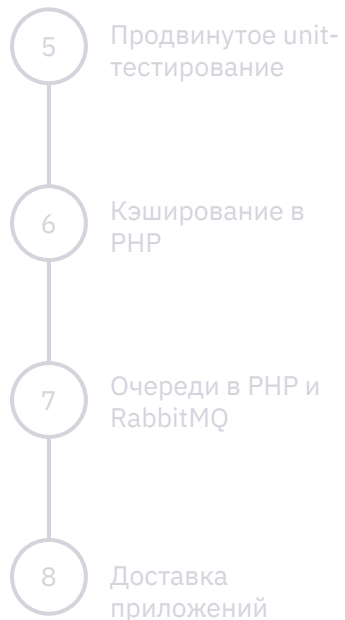
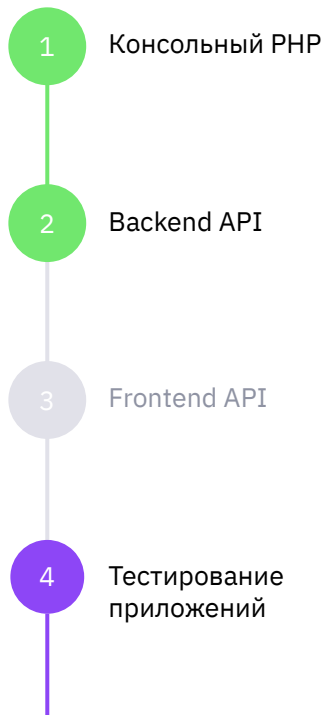
# Тестирование приложений

Урок 4








# План курса



WELCOME



## Что будет на уроке сегодня

-  Викторина, которая построена на основании реальных вопросов, которые задают на собеседовании
-  Имитация работы выполнения заданий от тимлида
-  Создание unit-теста на код, который мы написали ранее



# Викторина

Построена на основании реальных вопросов,  
которые задают на собеседовании

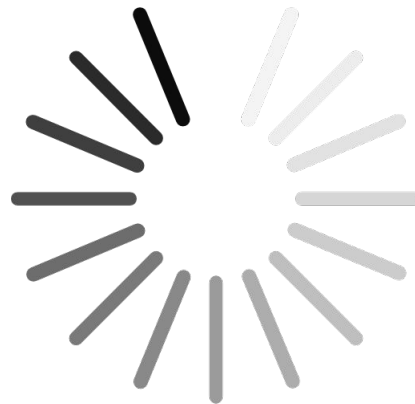


Преподаватель демонстрирует вопросы викторины, зачитывает их, а студенты пишут ответы в чате



## 1. Что такое тестирование приложений?

1. Стратегия тестирования, которая используется для проверки безопасности системы и анализа рисков, связанных с атаками хакеров
2. Тестирование уже проверенной функциональности после внесения изменений в код приложения
3. Тестирование, при котором выполняются тестовые сценарии, выявившие ошибки во время последнего запуска
4. Проверка и анализ приложения для выявления ошибок и дефектов, которые могут повлиять на его работу и надежность





## 2. У кого какая роль?

- 1) ? – Планирование по улучшению качества разработки и коммуникаций в команде
- 2) ? – Проводит тестирование приложений, чтобы выявить ошибки в работе приложения.
- 3) ? – Анализ результатов тестирования и качества новых версий продукта.







### 3. Чьи задачи:

- проверка готовности приложения к выпуску;
- оценка производительности и надежности приложения;
- в том числе задачи тестировщика (определение ошибок, сотрудничество с разработчиками, разработка тест-кейсов)?

1. Тестировщик

2. QC

3. QA





#### 4. Какие принципы тестирования вы знаете? Перечислите их!

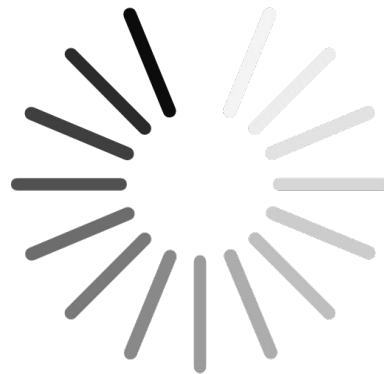
\*Вопрос без вариантов ответа\*



Подсказка: загляните в конспект



Совет: напишите ответ в чат или проговорите его

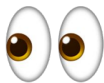




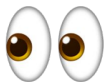


## 5. Перед вами классификации видов тестирования. Заполните недостающие названия:

статическое	?	динамическое
тестирование белого ящика	по доступу к коду и/или архитектуре	тестирование черного ящика тестирование серого ящика
автоматические	?	ручные
позитивное	по принципам работы с приложением	негативное
тестирование альфа-версии	?	тестирование бета-версии



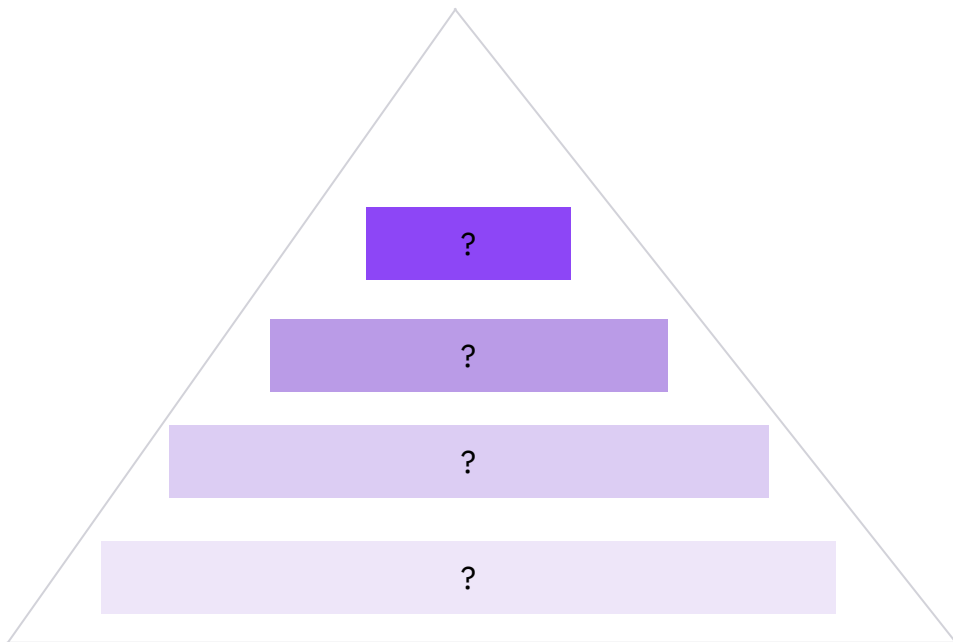
Подсказка: загляните в конспект



Совет: напишите ответ в чат или проговорите его



## 6. Перед вами уровни тестирования. Расположите уровни корректно:



Подсказка: загляните в конспект



Совет: напишите ответ в чат или проговорите его

1) Unit-тестирование

2) Функциональное

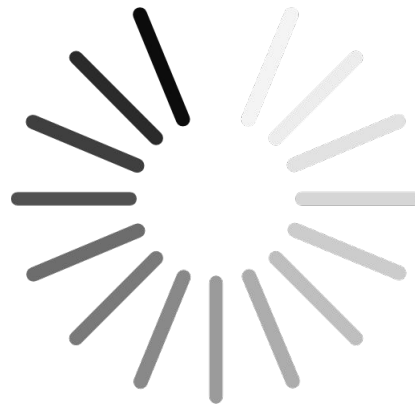
3) Приемочное

4) Системное



## 7. Что такое Unit-тестирование?

1. Процесс проверки отдельных компонентов приложения для убеждения в их правильной работе
2. Взаимодействие между разными частями приложения, чтобы найти проблемы и проверить, как работает система в целом
3. Проверка всей системы, используя реальные данные и ситуации, с которыми могут столкнуться пользователи





## 8. Для чего нам может понадобиться тестировать каждый компонент в отдельности?

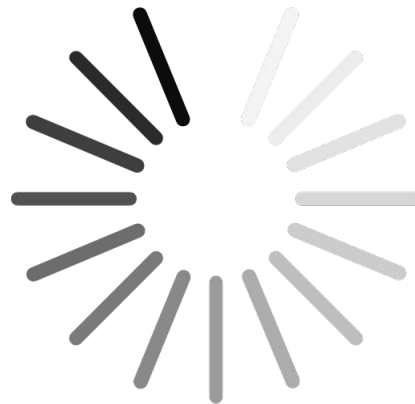
\*Вопрос без вариантов ответа\*



Подсказка: загляните в конспект



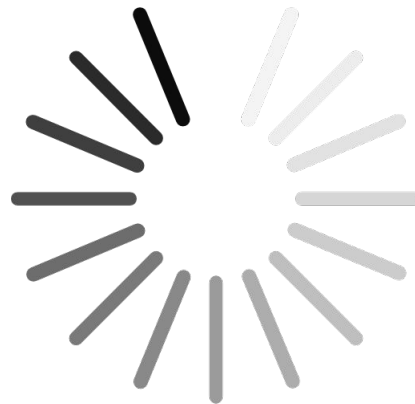
Совет: напишите ответ в чат или проговорите его





## 9. Что такое структура AAA в unit-тестах?

1. Шаблон организации тестовых случаев
2. Типы результатов, которые мы можем получить
3. Подход к разработке приложений, который помогает создавать качественный код и улучшать процесс разработки
4. Она проверяет, что условие истинно





## Перечень вопросов, которые также могут также задать:

1. Как вам доводилось тестировать приложения? Какие могут быть цели у тестирования?
2. Чем отличаются роли QA, QC и тестировщика? Какие у каждого задачи?
3. Какие классификации видов тестирования вы помните?
4. Какие уровни тестирования вы помните?
5. Как вы считаете, почему важность, сложность и стоимость выше всего у приемочного тестирования, а скорость запуска и количество ошибок выше у unit-тестирования?
6. Как создать unit-тест?
7. Что такое assert в unit-тестах? Какие из них вы помните?
8. Как мы можем передать тесту коллекцию значений? В чем ключевое отличие между TestWith и DataProvider?
9. Как запустить unit-тест? Какие типы результатов мы можем получить?
10. Что такое методологии тестирования BDD и TDD?



**Рекомендую на них ответить самостоятельно**  
**Ответы к ним вы найдете в лекции** 🙌



# Разбор домашнего задания





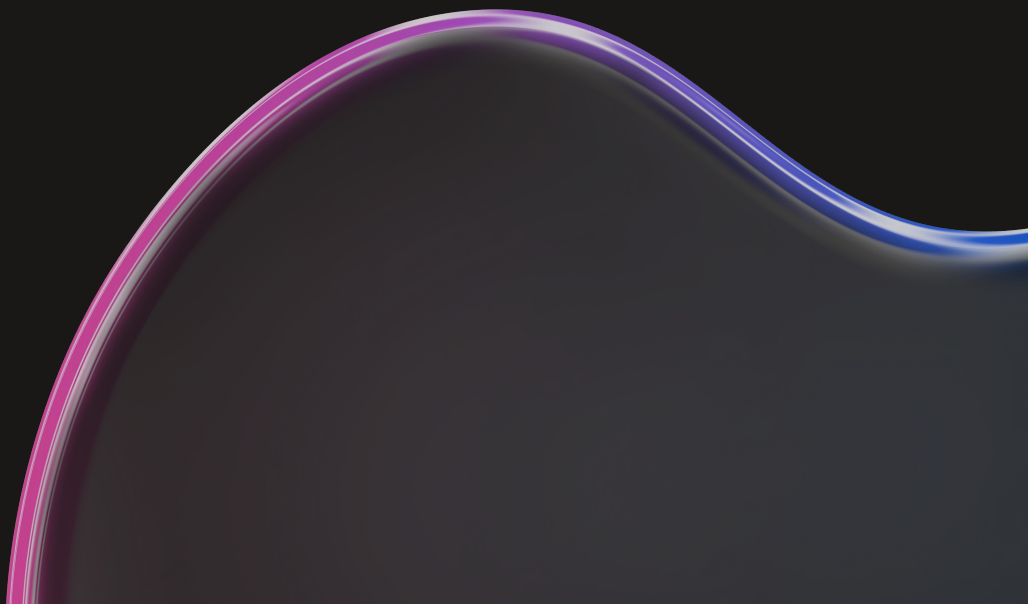


Сегодня мы работали с вами с токеном Telegram.  
Какой тип аутентификации мы использовали?





Кому удалось  
выполнить  
дополнительное  
задание № 4?





# Практика



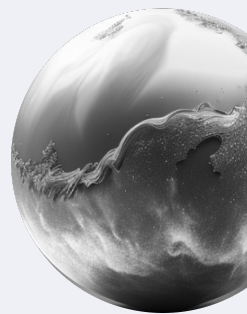


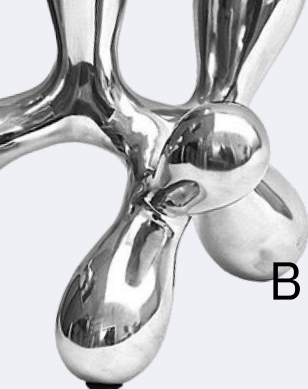
Тимлид в восторге от вашего приложения. Но, вот, проблема, нужно создать новый функционал, но ни одного unit-теста так и не было написано 😞

Оно и понятно, до этого мы создавали прототип, а сейчас переходим к серьезной разработке!

А как делать что-то новое или, что еще хуже, брать на проект еще одного разработчика без тестов?

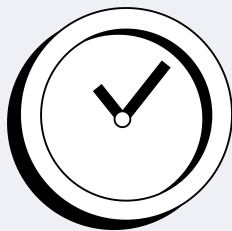
Срочно надо исправлять!



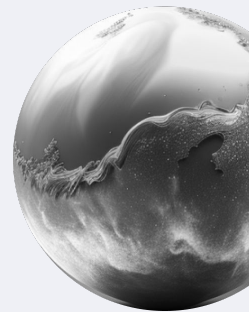


В этом задании определим, какие классы тестируем и какими сценариями.

Как минимум, один тест должен будет содержать параметризованные запросы, т.е. использовать **@testWith** или **DataProvider**



15 мин.





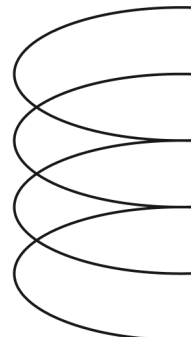
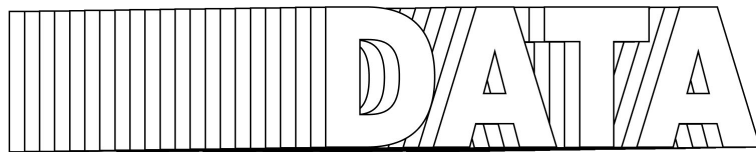
## Задание 2. Создание 4-5 unit-тестов по описанным сценариям

Разбейтесь на команды до 5 человек и напишите 4-5 тестов по указанным контрактам.

**1 тест должен содержатьDataProvider.**

Запустить тест, чтобы проверить, можно с помощью команды:

**php8.2 ./vendor/phpunit/phpunit/phpunit ./tests/**





# Домашнее задание







# Домашнее задание

## Задание

1. Сделайте новую ветку из той, которую мы создали на прошлом уроке. Это нужно для того, чтобы мы могли работать с кодом из прошлого урока.
2. Загрузите весь код из сегодняшнего урока в Git в новую ветку, создайте новый pull request. Пришлите на проверку ссылку на pull request.
3. ✖ Создайте тесты на оставшиеся контракты, которые мы придумали на уроке. Знаете еще сценарии, на которые можно написать unit-тест? Сделайте это! 🔥
4. ✖ Сломайте некоторые тесты и постарайтесь получить все типы результатов, которые мы разобрали в лекции.
5. ✖ На уроке мы рассмотрели много сценариев, на которые мы не можем создать тесты, т.к. они содержат побочные эффекты (соединяются с базой данных или с сервером telegram). Подумайте, как мы можем изменить код нашего приложения, чтобы создать unit-тест стало возможно.



На проверку отправляйте:

- ссылку на pull request в вашем репозитории с домашним заданием

✖ – дополнительное задание 💪







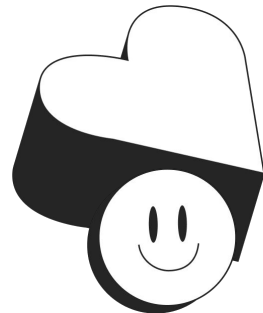
# Подведем итоги





## Подведение итогов

-  потренировались отвечать на вопросы про тестирование на собеседовании
-  научились работать в команде, понимать требования и выполнять задачи, поставленные тимлидом
-  научились разбивать задачу на более мелкие подзадачи, чтобы легче было управлять проектом и следить за прогрессом
-  сделали первый шаг в unit-тестировании





**Спасибо за внимание**