

## #7 Доверяй, но проверяй

В этом задании мы опишем поведение наших игровых объектов и попрактикуемся в использовании TDD.

### Задача

Используя подход TDD опишите функциональность, которая, воздействуя на структуры данных, управляет состоянием игры. Для управления структурами используйте чистые методы, которые работают только с входными данными, не производят сайд-эффектов и не изменяют исходных объектов (используют принцип **copy on write**).

Игра состоит из 10 вопросов

На всю игру отводится две минуты

Существуют два разных варианта вопроса:

- Выбор исполнителя из трех предложенных
- Выбор всех песен в названном жанре

Пользователь может ответить на вопрос правильно либо неправильно. Частично правильный ответ считается неправильным.

После ответа на все вопросы, пользователь переходит на экран статистики.

Пользователю дается право на три ошибки (три жизни). В случае неправильного ответа, пользователь переходит к следующему вопросу и у него снимается одна жизнь. Если были израсходованы все три жизни, пользователь переходит к экрану статистики.

### Правила формирования статистики

Статистика формируется относительно пользователей, сыгравших ранее. Сейчас не нужно реализовывать запись статистики предыдущих игр, поэтому заведите отдельную структуру данных, которая будет хранить тестовые результаты.

На экране статистики показывается затраченное время, количество правильных ответов и процент игроков, лучше которых сыграл пользователь. Фактически, процент зависит от позиции игры в списке прошедших игр, отсортированных по убыванию количества правильных ответов и возрастанию времени.

Например, последняя игра попала в статистику второй, со временем 32 и 10 правильными ответами:

```
const statistics = [  
  { time: 20, answers: 10 },  
  { time: 32, answers: 10 },  
  { time: 44, answers: 10 },  
  { time: 20, answers: 8 },  
  { time: 50, answers: 7 }  
]
```

В этом случае результат лучше, чем у 60% игроков, потому что игра находится на 2 месте из 5, а это лучше чем у троих игроков из 5:  $3/5 = 0.6 = 60\%$ .

### **Обратите внимание**

Все функции, которые изменяют объект состояния игры — должны быть «чистыми», т.е. принимать на вход исходное состояние игры и отдавать новое состояние игры с измененным полем, в соответствии с тем, какое поле изменилось.

---