Раздел. **Ручное тестирование**

Модуль. **Техники тест-дизайна**

Автор. **Константин Фирсанов**

**Примеры и решения**

Классы эквивалентности и таблицы решений / альтернатив

Пример № 1

Тестируем функциональность приложения, позволяющего покупать авиа- и железнодорожные билеты онлайн. Стоимость билета будет зависеть от возраста пассажира, так как дети, студенты и пенсионеры относятся ко льготным категориям.

У нас есть четыре возрастных группы: младше 15 лет, от 15 до 25 лет, старше 25 и младше 60 лет и люди старше 60. При этом, в поле для ввода возраста помещается всего два символа, поэтому указать возраст более 99 лет технически невозможно.

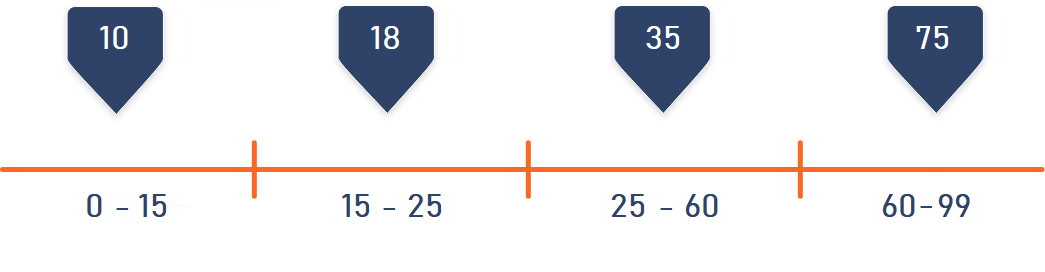
**Класс 1** младше 15 лет (дети)

**Класс 2** от 15 до 25 лет (студенты по акции)

**Класс 3** от 25 до 60 лет (обычный тариф)

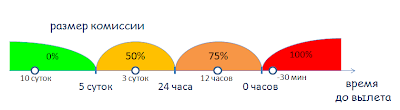
**Класс 4** старше 60 лет (пенсионный тариф)

Можно провести еще один тест для случая, если возраст человека превышает 99 лет. Да, последний тест на практике невыполним (поскольку в поле возраста невозможно ввести более двух знаков), и все же не следует забывать об этой проверке.



Пример № 2

Давайте рассмотрим пример: функцию подсчета комиссии при отмене бронирования авиабилетов.



Предположим, что размер комиссии зависит от времени до вылета, когда совершена отмена:

* За 5 суток до вылета комиссия составляет 0%
* Меньше 5 суток, но больше 24 часов – 50%
* Меньше 24 часов, но до вылета – 75%
* После вылета – 100%

Теперь давайте пойдем по шагам:

**1. Определим классы эквивалентности** (для каждого теста из этих классов мы ожидаем получить одинаковый результат):

* 1 класс: время до вылета > 5 суток
* 2 класс: 24 часа < время до вылета < 5 суток
* 3 класс: 0 часов < время до вылета < 24 часа
* 4 класс: время до вылета < 0 часов (вылет уже состоялся)

**2. Выберем представителя от каждого класса**. Здесь мы можем поступить, как нам хочется, и выбрать любые значения из класса. Ведь, если предположить, что мы правильно разбили на классы эквивалентности, то нет разницы, какое значение из диапазона мы выберем.

* время до вылета = 10 суток (тест из 1-го класса)
* время до вылета = 3 суток (тест из 2-го класса)
* время до вылета = 12 часов (тест из 3-го класса)
* время до вылета = -30 мин (тест из 4-го класса)

**3. Выполним тесты**:

* Отменим бронь за 10 суток до вылета и проверим, что комиссия составила 0%.
* Отменим бронь за 3 суток до вылета и проверим, что комиссия составила 50%.
* Отменим бронь за 12 часов до вылета и проверим, что комиссия составила 75%.
* Отменим бронь через 30 мин после вылета и проверим, что комиссия составила 100%.

Мы видим, что у нас осталось всего 4 теста. А сколько возможных тестов существует?

Даже если мы введем ограничение, что отмена бронирования может произойти в рамках 10 суток до вылета и 1 суток после вылета, то у нас будет около 950400 возможных тестов (просчитано количество секунд в 11 сутках).

Техники “таблица решений”

Пример № 3

Рассмотрим таблицу принятия решений на примере страницы регистрации нового пользователя сервиса KUKU.io

Используем понятия “корректные” и “некорректные” данные.

Чтобы регистрация прошла успешно, необходимо заполнить корректными данными оба поля. Если поля заполняются некорректными данными, то система должна выдать ошибку: “Введены невалидные данные”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Условие** | **Значение 1** | **Значение 2** | **Значение 3** | **Значение 4** |
| Ввод корректных данных в поле e-mail | **+** | **-** | **+** | **-** |
| Ввод корректных данных в поле password | **+** | **-** | **-** | **+** |
| Ввод некорректных данных в поле e-mail | **-** | **+** | **-** | **+** |
| Ввод некорректных данных в поле password | **-** | **+** | **+** | **-** |
| **Действия** |  |  |  |  |
| Регистрация прошла успешно |  | **-** | **-** | **-** |
| Выдается ошибка: “Введены невалидные данные” | **-** | **+** | **+** | **+** |

Значения 2, 3, 4 приводят к одному и тому же результату с разными входными значениями.

Пример № 4

Предположим, чтобы войти в систему, пользователю нужно ввести сначала логин и пароль, а затем еще подтвердить свою личность присланным в смс кодом.

Какие возможны сценарии:

1.       Правильный логин и правильный пароль.

2.       Правильный логин, неправильный пароль.

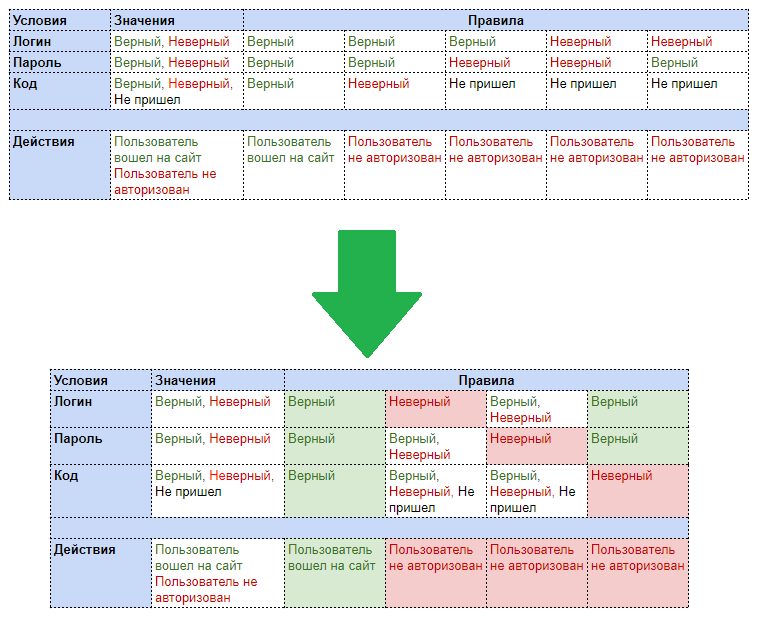
3.       Неправильный логин, правильный пароль.

4.       Неправильный логин, неправильный пароль.

Первый из этих сценариев сопровождается либо правильным, либо неправильным вводом смс-кода, итого у нас получается 5 тестов. При этом только один из сценариев приведет к положительному результату (пользователь успешно авторизуется), а остальные закончатся неудачей.

Однако, может быть так, что система выдает разные сообщения в зависимости от того, на каком этапе была допущена ошибка, скажем: invalid login, invalid password. Соответственно, групп потребуется больше, а таблица станет обширнее.

Этот метод хорош тем, что он показывает сразу все возможные сценарии в форме, понятной даже неспециалисту.



*Пример таблицы принятия решений*