

Тест-дизайн и тест-аналитика
Урок 2

Классы ЭКВИВАЛЕНТНОСТИ





Оглавление

На этом уроке	3
Что такое тест-дизайн, и зачем он нужен?	3
Классы эквивалентности	4
Пример: уровень шума домашних животных	4
Пример: найм на работу	5
Алгоритм работы с классами эквивалентности	7
Пример 1: отмена авиабилетов	7
Пример 2: знаки зодиака	9
Пример 3: загрузка фото	11
Контрольные вопросы	12
Глоссарий	13
Дополнительные материалы	13



На этом уроке

1. Повторим, что такое тест-дизайн и его назначение.
2. Рассмотрим классы эквивалентности и алгоритм и примеры работы с ними.

Что такое тест-дизайн, и зачем он нужен?

Тест-дизайн — это этап тестирования ПО, на котором проектируют и создают тест-кейсы.

Допустим, есть задача: протестировать простейшую форму регистрации с двумя полями: логином и паролем. Сколько тестов нужно, чтобы установить уровень качества функциональности?

Можно предположить, что два:

- позитивный сценарий (логин test, пароль test1!);
- негативный сценарий (логин и пароль не заполнены).

Но что будет, если:

- логин уже занят;
- в логине есть спецсимволы */%;
- в логине есть пробелы;
- пароль состоит из одного символа.

А сколько ещё внештатных ситуаций может быть? Как покрыть их тест-кейсами?

Вспомним один из принципов тестирования:

Исчерпывающее тестирование невозможно.

Полное тестирование с использованием всех комбинаций вводов и предусловий невыполнимо, кроме тривиальных случаев. Вместо исчерпывающего тестирования используется анализ рисков и расстановка приоритетов, чтобы точнее сфокусировать усилия по тестированию.

При этом надо протестировать достаточно хорошо. Поэтому есть правила составления тестов и подбора тестовых данных. Они вырабатывались на протяжении всего существования IT-отрасли и сейчас представляют собой набор



лучших практик, которых придерживается хороший тестировщик. Их соблюдение гарантирует оптимальный охват тестируемой функциональности, при котором большая часть дефектов будет обнаружена и не дойдёт до пользователя. Такие правила называются **техниками тест-дизайна**.

Распространённые техники:

- Классы эквивалентности (эквивалентное разделение).
- Граничные значения (анализ граничных значений, метод граничных значений).
- Попарное тестирование (тестовая комбинаторика, Pairwise).
- Тестирование состояний и переходов.
- Таблицы принятия решений.
- Исследовательское тестирование.

Техники тест-дизайна позволяют значительно сократить количество тестов. Они концентрируются на уязвимых и важных участках функционала, не жертвуя при этом качеством.

На этом уроке мы разберёмся с первой техникой — классами эквивалентности.

Классы эквивалентности

Класс эквивалентности — набор входных значений, каждое из которых одинаково обрабатывается и приводит к одному результату. У значений внутри класса одни признаки, что приводит к их идентичной обработке.

Пример: уровень шума домашних животных

Рассмотрим классы эквивалентности на примере домашних животных — кошек и попугаев. Конечно, мир природы разнообразнее, чем программное обеспечение, поэтому будем упрощать.

У всех кошек есть общие признаки:

- мурчат, когда их чешут за ушком;
- любят охотиться;
- делают тыгыдык по ночам.

У попугаев другие общие признаки:

- умеют летать;



- могут научиться разговаривать;
- умеют петь.

Все кошки составляют один класс эквивалентности, а попугаи — другой. Любая кошка будет мурчать, но не сможет летать. Любой попугай может петь, но не охотится по ночам.

Допустим, перед нами стоит задача измерить уровень шума от разных домашних животных. Зная, что все кошки и все попугаи производят примерно одинаковые звуки, нам не придётся исследовать всех кошек и всех попугаев на планете. Достаточно взять одну любую кошку и одного любого попугая, чтобы измерить их уровень шума и с большой долей вероятности получить корректные результаты.

Пример: найм на работу

В требованиях о найме у HR-отдела есть условие, которое автоматически распределяет резюме кандидатов в разные категории:

Возраст кандидата, лет	Статус резюме
0-15	Не нанимать, NO
16-17	Сокращённый рабочий день, максимум 4 часа, PART
18-64	Полный рабочий день, максимум 8 часов, FULL
65-99	Не нанимать, NO

Посмотрим, как это могло бы выглядеть в коде приложения. Обратите внимание на знаки `>=` и `<=`. Они важны при определении класса эквивалентности и граничных значений.

```
If (applicantAge >= 0 && applicantAge <16)
    hireStatus="NO";
If (applicantAge >= 16 && applicantAge <18)
    hireStatus="PART";
If (applicantAge >= 18 && applicantAge <65)
    hireStatus="FULL";
If (applicantAge >= 65 && applicantAge <=99)
    hireStatus="NO";
```

Если проводить исчерпывающее тестирование и проверять все варианты, количество тестов составит минимум 100 (без учёта проверок отрицательных



значений, значений больше 99, символов, пустого ввода и прочего). Выполнить их невозможно, поэтому нужно:

1. Разделить данные на классы эквивалентности.
2. Выбрать хотя бы одно значение из каждого класса эквивалентности для проверки.

Получаем следующие проверки:

- 1-й класс эквивалентности — 0 лет;
- 2-й класс эквивалентности — 16 лет;
- 3-й класс эквивалентности — 18 лет;
- 4-й класс эквивалентности — 65 лет.

Почему лучше выбрать именно эти значения, разберём позже, когда речь пойдёт о граничных значениях. Пока обращаем внимание на то, что каждое из выбранных значений находится в указанном классе.

Из каждого класса эквивалентности мы выбрали значения, чтобы сократить количество тестов (теперь их 4 вместо 100). Любые другие значения из класса эквивалентности должны давать те же результаты, что и выбранные.

Если известно, что есть группа данных, использование которых приводит систему в одно и то же состояние,
нет необходимости проверять каждое значение из этой группы отдельно.

Исключения возможны, но мы не можем проверять все данные, поэтому приходится прибегать к подобным допущениям.

Выделяют два вида классов эквивалентности:

- **линейные** (упорядоченные) — значения можно упорядочить и расположить на шкале. У линейных классов эквивалентности всегда есть границы, где заканчивается один класс и начинается другой. Границы могут входить в класс, а могут быть самостоятельными единицами.
- **нелинейные** (неупорядоченные) — значения нельзя упорядочить, граничных значений нет.

Тестирование на основе классов эквивалентности — техника тест-дизайна на основе метода чёрного ящика, в ходе которой все данные делятся на классы



эквивалентности, и затем тестирование проводится на одном значении из этого класса.

Тестирование на основе классов эквивалентности основано на двух умозаключениях:

1. Если одно значение из класса выявит ошибку, остальные, скорее всего, тоже это сделают.
2. Если одно значение из класса не выявит ошибку, остальные, скорее всего, тоже этого не сделают.

Исчерпывающее тестирование невозможно,
и есть вероятность, что отдельные значения в классе
эквивалентности поведут себя не так, как остальные.

Поэтому в описаниях часто встречаются комментарии
«скорее всего», «с большой долей вероятности».

На классы эквивалентности можно разбить:

- **символы** — они могут быть валидными (@ в адресе электронной почты) и невалидными (?, %, *);
- **длину строки** — например, валидный класс от 1 до 30 знаков, невалидный — всё остальное (меньше 1 и больше 30);
- **объём памяти**, который необходим приложению для стабильной работы;
- **разрешение экрана** — всё, что меньше или больше заявленных требований к разрешению экрана, будет относиться к невалидным классам;
- **версии операционных систем, библиотек** — также определяются согласно требованиям. Например, приложение должно работать на ОС Windows 7, но поддержка Windows Nt уже не требуется.
- **объём передаваемых данных** — по требованиям. Например, если мощности сервера не позволяют обработать объём данных больше определённого значения.

Алгоритм работы с классами эквивалентности

1. На основе анализа нужно выбрать параметры, которые влияют на результат.
2. Для каждого параметра выделить классы эквивалентности.
3. Из каждого класса эквивалентности выбрать одно значение.

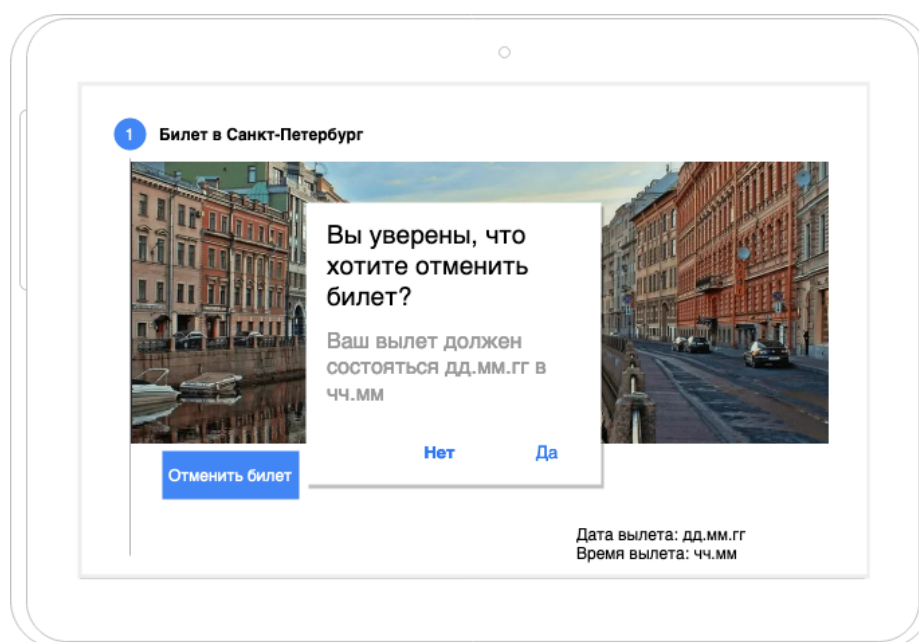


4. Обработать выбранные значение в соответствии с pairwise при необходимости — об этой технике поговорим в следующий раз.

Рассмотрим два примера тестирования на основе классов эквивалентности.

Пример 1: отмена авиабилетов

Требование на фичу «Отмена авиабилетов»



Подсчёт комиссии при отмене бронирования авиабилетов производится в зависимости от времени отмены на следующих условиях:

- за 5 суток до вылета комиссия составляет 0% от стоимости билета;
- меньше 5 суток, но больше 24 часов — 50% от стоимости билета;
- меньше 24 часов, но до вылета рейса — 75% от стоимости билета;
- после вылета рейса — 100% от стоимости билета.

Сначала определим классы эквивалентности:

- 1-й класс эквивалентности: время до вылета > 5 суток;
- 2-й класс эквивалентности: 5 суток => время до вылета > 24 часа;
- 3-й класс эквивалентности: 24 часа => время до вылета > 0 часов;
- 4-й класс эквивалентности: время до вылета < 0 часов (рейс уже вылетел).

Затем из каждого класса эквивалентности выберем одно значение:

- 1-й класс эквивалентности = 10 суток до вылета;



- 2-й класс эквивалентности = 3 суток до вылета;
- 3-й класс эквивалентности = 12 часов до вылета;
- 4-й класс эквивалентности = 30 минут после вылета.

В результате получим следующие тесты:

№	Проверка	Ожидаемый результат
1	Отменить бронь авиабилета за 10 суток до вылета	Размер комиссии — 0% от стоимости
2	Отменить бронь авиабилета за 3 суток до вылета	Размер комиссии — 50% от стоимости
3	Отменить бронь авиабилета за 12 часов до вылета	Размер комиссии — 75% от стоимости
4	Отменить бронь авиабилета через 30 минут после вылета	Размер комиссии — 100% от стоимости

В примерах выше были данные, которые можно расположить на числовой прямой — их классы эквивалентности будут линейными. Такие классы можно разбить на диапазоны с точными границами начала и конца (от 0 до 15, от 16 до 18 и далее).

Есть и нелинейные классы эквивалентности — это набор неупорядоченных данных, которые не имеют границ и являются частью множества данных. Пример — расширения файлов, операционные системы, группы пользователей с различными правами (например, пользователь, модератор, администратор) и так далее. В этом случае можно выделить только два класса эквивалентности:

- **валидный** — соответствует требованиям,
- **невалидный** — не соответствует требованиям или обрабатывается системой отличным от валидного класса образом.

Пример 2: знаки зодиака

User Story: Я как пользователь хочу определить свой знак зодиака по дате рождения.



Узнайте свой знак Зодиака!

День	Месяц	Год
<input type="text"/>	<input type="text"/>	<input type="text"/>

Узнать!

Макет будущего сайта

Пользовательский сценарий:

Пользователь: заходит на страницу

Пользователь: заполняет день, месяц и год рождения

Пользователь: нажимает кнопку «Узнать»

Система: проверяет, что значение поля «Месяц» от 1 до 12

Система: проверяет значение поля «День»

Система: проверяет значение поля «Год» — високосный или нет

- от 1 до 28, если Месяц = 2 и год не високосный
- от 1 до 29, если Месяц = 2 и год високосный
- от 1 до 30, если Месяц = 4, 6, 9, 11
- от 1 до 31, если Месяц = 1, 3, 5, 7, 8, 10, 12

Дата корректная

Система: показывает знак зодиака, соответствующий дате.

Дата некорректная

Система: показывает сообщение «Указанная дата не существует»

Естественно, мы не будем тратить время и проверять каждый день в году, а воспользуемся техникой классов эквивалентности. Сначала рассмотрим позитивный сценарий, когда пользователь ввёл существующую валидную дату (например, 31.03.2000). Код конвертирует дату в порядковый номер дня в году.

Какие классы эквивалентности можно выделить? Поскольку знаков зодиака 12, то и классов эквивалентности будет столько же — по одному на каждый знак. Присвоим каждому дню года порядковый номер, где 1 — это 1 января, а 365 — это 31 декабря (рассмотрим невисокосный год). Получим следующие классы:



- 1) 1-20 — козерог
- 2) 21-50 — водолей
- 3) 51-79 — рыбы
- 4) 80-110 — овен
- 5) 111-141 — телец
- 6) 142-172 — близнецы
- 7) 173-203 — рак
- 8) 204-233 — лев
- 9) 234-266 — дева
- 10) 267-296 — весы
- 11) 297-326 — скорпион
- 12) 327-356 — стрелец
- 13) 357-365 — козерог

Из-за того, что гороскоп не совпадает с календарным годом, получилось не 12, а 13 классов. Теперь для тестирования достаточно взять по одному значению из каждого интервала и проверить его. Например, если мы укажем 30-й день года (30 января), то результат будет «водолей». Получается, для минимального тестового покрытия нам понадобится 13 тестов.

Рассмотрим альтернативный сценарий — пользователь ввёл невалидную дату. Ожидаем получить сообщение о неверно введенных датах или (в случае с несуществующими 32 числом и 14 месяцем) невозможность ввода. Возможны несколько вариантов:

- 29.02.2001 (год невисокосный, 29 февраля в нём нет)
- 32.03.2000 (32 числа нет ни в одном месяце)
- 31.04.1999 (31 число бывает, но не в апреле)
- 25.14.1995 (нет 14 месяца)
- 00.00.0000 (если речь о цифрах, нули проверяем с особым пристрастием)

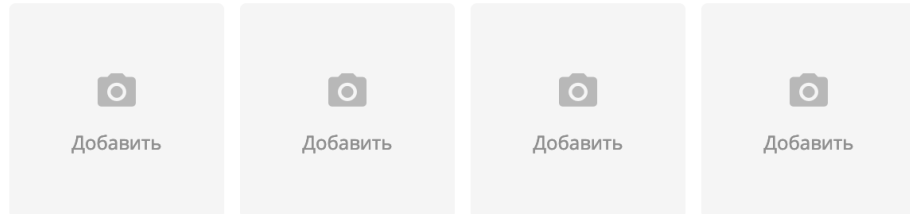
Можно проявить воображение и придумать ещё больше негативных проверок. Таким образом, для тестирования простейшей странички нам понадобится минимум 18 тестов. Так как эти классы эквивалентности линейны, у них есть границы. Это элементы повышенного риска, и их тестирование мы рассмотрим в теме «Доменный анализ». Это ещё немного увеличит наш список необходимых проверок.



Пример 3: загрузка фото

Фотографии*

Перетащите фото сюда или [выберите их на своем компьютере...](#)



Первое фото будет отображаться в результатах поиска, выберите наиболее удачное.
Вы можете загрузить до 10 фотографий в формате JPG или PNG.
Максимальный размер фото — 25MB.

На скриншоте видим — валидными форматами будут только JPG и PNG, можно загрузить до 10 фото, максимальный размер фото — 25 MB. Определим параметры и классы эквивалентности для каждого из них.

Получается три параметра:

- формат изображения;
- количество загружаемых фото;
- размер фото.

Составим для них классы эквивалентности. Обратите внимание, что фотографии — обязательное поле (есть звёздочка), поэтому мы не можем оставить его пустым.

Также учтём, что количество фото и размер не могут быть отрицательными.

Формат	Количество фото	Размер фото, МБ
JPG	[1; 10]	(0; 25]
PNG	Больше 10	Больше 25
другое	0	0

Теперь выделим среди классов невалидные — то есть те, из-за которых будет невозможно загрузить фото:

Формат	Количество фото	Размер фото
JPG	[1; 10]	(0; 25]
PNG	Больше 10	Больше 25



другое	0	0
--------	---	---

Для тестирования достаточно взять по одному значению из каждого класса: получим 4 позитивных и 5 негативных проверок.

Контрольные вопросы

1. Какие бывают классы эквивалентности?
2. С какими классами эквивалентности вы сталкиваетесь в жизни? Приведите примеры.
3. Что можно разбить на классы эквивалентности?

Глоссарий

Тест-дизайн — этап тестирования ПО, на котором проектируют и создают тест-кейсы.

Класс эквивалентности — набор входных значений, каждое из которых обрабатывается одинаково и приводит к одному результату.

Дополнительные материалы

1. [Классы эквивалентности для имен](#)
2. [Немного о простом. Тест-дизайн](#)