

Классы ЭКВИВАЛЕНТНОСТИ

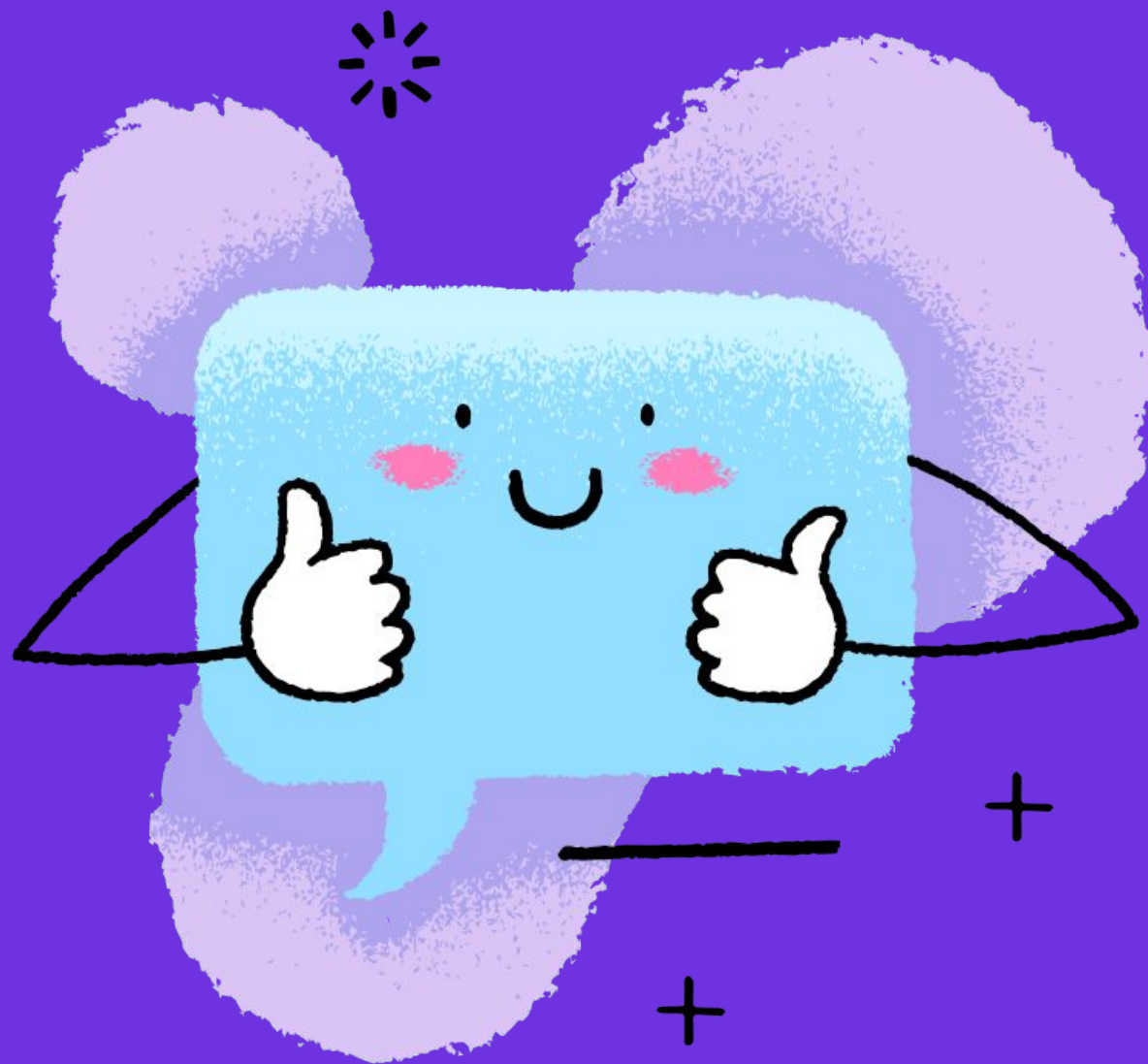
Цели урока

1. Узнаем, что такое техники тест-дизайна и зачем они нужны
2. Рассмотрим классы эквивалентности
3. Узнаем, как работает pairwise
4. Составим тестовые наборы данных по принципу pairwise с использованием PICT
5. Рассмотрим pairwise и негативное тестирование

План урока

1. Что такое тест-дизайн
2. Классы эквивалентности
3. Попарное тестирование - pairwise

Тест-дизайн



Тест-дизайн

Тест-дизайн — это этап тестирования ПО, на котором проектируются и создаются тест-кейсы.

Тест-дизайн

Задача: протестировать форму регистрации с 2 полями: логин и пароль.
Сколько тестов нужно, чтобы установить уровень качества?

Интуитивно можно предположить:

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

Тест-дизайн

Но что будет, если:

- такой логин уже занят?
- логин содержит спецсимволы */% и т.д.?
- логин с пробелами?
- пароль из 1 символа

А сколько еще внештатных ситуаций может быть? И как покрыть их все тест-кейсами?

Техники тест-дизайна обеспечивают оптимальное тестовое покрытие при ограниченном количестве проверок

Популярные техники тест-дизайна

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

Классы эквивалентности

теория



Классы эквивалентности

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

Значения внутри класса обладают общими признаками, что и приводит к идентичной обработке.

Классы эквивалентности

Класс “Кошки”

- мурлыкают
- охотятся
- делают “тыгыдык”

Класс “Попугаи”

- летают
- учатся разговаривать
- поют

Все кошки составляют один класс эквивалентности, а попугайчики - другой класс.

Допустим, перед нами стоит задача измерить уровень шума от разных домашних животных. Достаточно взять любую кошку и любого попугайчика, измерить их уровень шума, и мы с большой долей вероятности получим корректные результаты.

Классы эквивалентности

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

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

Классы эквивалентности

Линейные (упорядоченные)

- значения можно упорядочить и расположить на шкале;
- есть границы, где заканчивается 1 класс и начинается другой

Нелинейные (неупорядоченные)

- значения нельзя упорядочить;
- граничных значений нет

Классы эквивалентности

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.

Классы эквивалентности

- символы
- длину строки
- объём памяти
- разрешение экрана
- версии операционных систем, библиотек
- объём передаваемых данных

Классы эквивалентности - алгоритм

1. на основе анализа выбрать параметры, которые влияют на результат
2. для каждого параметра выделить классы эквивалентности
3. из каждого класса эквивалентности выбрать 1 значение
4. обработать выбранные значение в соответствии с pairwise

Классы эквивалентности

примеры



Классы эквивалентности

пример “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"/>

Use Case

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”

Классы эквивалентности

пример “Знаки зодиака”

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

Итого необходимо 13 позитивных проверок, чтобы убедиться в правильности работы определителя.

Классы эквивалентности

пример “Знаки зодиака”

Негативные проверки:

- 29.02.2001 (год невисокосный, 29 февраля в нем 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 идет о цифрах, нули проверяем с особым пристрастием)

Классы эквивалентности

пример “Прогноз погоды”

User story: Я как пользователь хочу узнать прогноз погоды в указанном городе с нужной степенью детализации

Прогноз погоды

Город	▼
-------	---

RU	US
----	----

Осадки	<input type="radio"/>
--------	-----------------------

По дням	По часам
---------	----------

Use Case

Пользователь: заходит на сайт

Пользователь: выбирает город из списка

Пользователь: выбирает язык (по умолчанию выбран русский)

Пользователь: выбирает детализацию по осадкам (по умолчанию выключено)

Пользователь: выбирает детализацию по времени (по умолчанию по дням)

Система: показывает пользователю прогноз в соответствии с указанными значениями

Используется 3 алгоритма расчета прогноза:

- для городов федерального значения (Москва, Санкт-Петербург, Севастополь)
- для городов областного, республиканского, краевого, окружного значения (Волгоград, Казань, Краснодар и т.д.)
- для городов районного значения (Бологое, Суоярви т. д.)

Классы эквивалентности

пример “Прогноз погоды”

Город	Язык	Осадки	Детализация
Федерального значения	Русский	Да	По дням
Областного значения	Английский	Нет	По часам
Районного значения			

Классы эквивалентности

пример “Прогноз погоды”

Большинство дефектов возникают при комбинации только **двух параметров**. Есть смысл проверять все возможные сочетания города и языка, города и осадков, города и детализации, т.е. проверять параметры **попарно**.

Эта техника называется **попарным тестированием**, или **pairwise**.

Классы эквивалентности

пример “Прогноз погоды”

Начнем с пары “Город - Язык”, а остальные ячейки пока что заполнять не будем

	Город	Язык	Осадки	Детализация
1	Москва	Русский		
2	Москва	Английский		
3	Волгоград	Русский		
4	Волгоград	Английский		
5	Бологое	Русский		
6	Бологое	Английский		

Классы эквивалентности

пример “Прогноз погоды”

Заполним “Осадки”, чтобы перебрать все возможные пары “Город - Осадки”

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Да	
2	Москва	Английский	Нет	
3	Волгоград	Русский	Да	
4	Волгоград	Английский	Нет	
5	Бологое	Русский	Да	
6	Бологое	Английский	Нет	

Классы эквивалентности

пример “Прогноз погоды”

Проверим пару “Язык - Осадки”. Не хватает пары “Русский - Нет”, “Английский - Да”. Поменяем местами значения осадков в строке 1 и 2.

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	
2	Москва	Английский	Да	
3	Волгоград	Русский	Да	
4	Волгоград	Английский	Нет	
5	Бологое	Русский	Да	
6	Бологое	Английский	Нет	

Классы эквивалентности

пример “Прогноз погоды”

Аналогично заполняем колонку “Детализация”. Сопоставляем ее с городом, а затем проверяем, чтобы были все пары “Язык - Детализация”, “Осадки - Детализация”

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	По дням
2	Москва	Английский	Да	По часам
3	Волгоград	Русский	Да	По часам
4	Волгоград	Английский	Нет	По дням
5	Бологое	Русский	Да	По дням
6	Бологое	Английский	Нет	По часам

Классы эквивалентности

пример “Прогноз погоды”

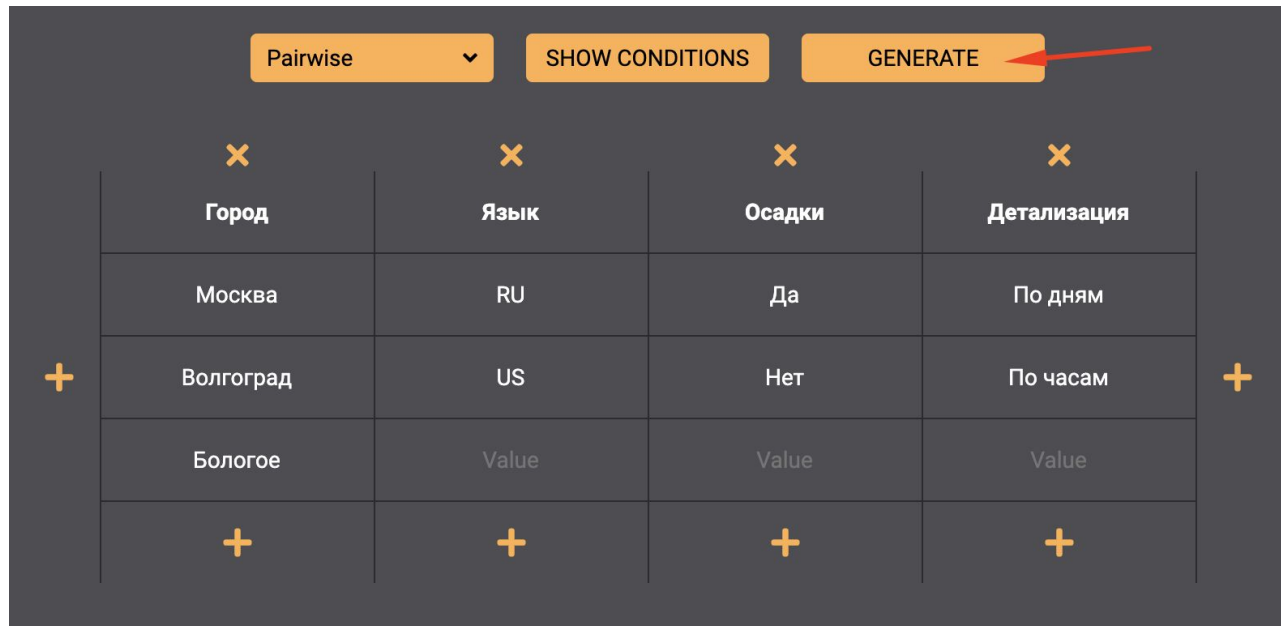
Вместо 24 тестов осталось всего 6, и это в 4 раза сократит время на тестирование.

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	По дням
2	Москва	Английский	Да	По часам
3	Волгоград	Русский	Да	По часам
4	Волгоград	Английский	Нет	По дням
5	Бологое	Русский	Да	По дням
6	Бологое	Английский	Нет	По часам

PICT

Существуют специальные программы, которые сами составляют таблицы из исходных параметров:

- <https://slothman.dev/pairwise-generator/>
- <https://pairwise.yuuniworks.com/>



The screenshot shows a web interface for a pairwise generator. At the top, there is a dropdown menu set to 'Pairwise', a 'SHOW CONDITIONS' button, and a 'GENERATE' button with a red arrow pointing to it. Below these is a table with four columns: 'Город' (City), 'Язык' (Language), 'Осадки' (Precipitation), and 'Детализация' (Detailing). The table has five rows. The first row contains 'Москва' (Moscow), 'RU', 'Да' (Yes), and 'По дням' (By day). The second row contains 'Волгоград' (Volgograd), 'US', 'Нет' (No), and 'По часам' (By hour). The third row contains 'Бологое' (Bologoye), 'Value', 'Value', and 'Value'. The fourth row is empty. The table is flanked by a large orange '+' on the left and a large orange 'x' on the right. Each column header has an orange 'x' above it, and each row has an orange '+' at the bottom.

Город	Язык	Осадки	Детализация
Москва	RU	Да	По дням
Волгоград	US	Нет	По часам
Бологое	Value	Value	Value

Pairwise и негативное тестирование

Составляя тестовые наборы для попарного тестирования, никогда не учитываем негативные сценарии, т.е. те, которые способны привести к ошибке.

Невалидные значения **никогда не комбинируются друг с другом**, т.к. если в тесте будет 2 или более невалидных параметра, то тестировщик не установит причину возникновения ошибки.

Pairwise и негативное тестирование

Проверки с невалидными данными добавляются в таблицу по принципу “1 невалидный параметр в строке”. Остальные параметры могут быть любыми валидными.

	Город	Язык	Осадки	Детализация
1	Москва	Русский	Нет	По дням
2	Москва	Английский	Да	По часам
3	Волгоград	Русский	Да	По часам
4	Волгоград	Английский	Нет	По дням
5	Бологое	Русский	Да	По дням
6	Бологое	Английский	Нет	По часам
7	Не заполнено	Английский	Нет	По часам
8	Бологое	Китайский	Да	Нет

Спасибо!
Каждый день
вы становитесь
лучше :)

