

Курс Тестировщик ПО

Блок4 Тест-дизайн

бруноям



Структура:

- Тест-анализ
- Тестирование без документации
- Тестовая документация
- Техники тест-дизайна
- Проверки типов данных
- Пример тестирования формы

Тест-анализ:

Тест-анализ - процесс поиска и рассмотрения информации, необходимой для тестирования.

Вопросы тест-анализа:

1. Из чего состоит продукт?
2. Что он умеет?
3. Какой он?
4. Как взаимодействует с внешней средой? Системы и пользователи.

Тест-анализ:

Декомпозиция - разделение целого на части.

Варианты декомпозиции:

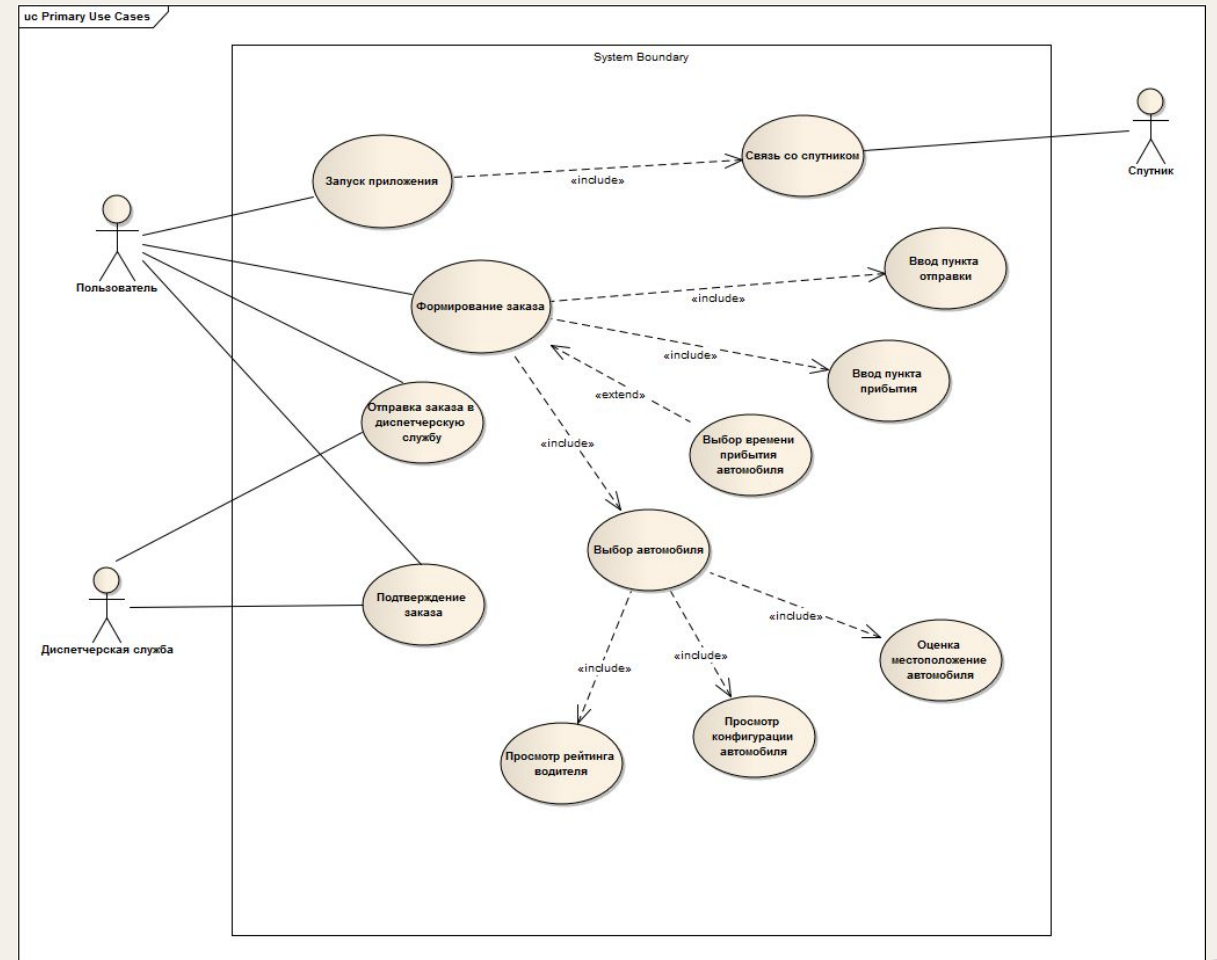
- По функциональности
- По страницам
- По тестовым данным
- По модулям

Тест-анализ:

Сценарии использования (Use Cases).

UML (Unified Modeling Language)

- унифицированный язык моделирования.
Язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.



Use-case в виде таблицы:

Действующие лица	Пользователь, Система
Цели	Пользователь: авторизоваться в системе и начать работать; Система: идентифицировать пользователя и его права.
Успешный сценарий: <ol style="list-style-type: none">1. Пользователь запускает систему. Система открывает сессию пользователя, предлагает ввести логин и пароль.2. Пользователь вводит логин и пароль.3. Система проверяет логин и пароль.4. Система создает запись в истории авторизаций (IP адрес пользователя, логин, дата, рабочая станция).5. Система выдает пользователю сообщение по поводу успешной авторизации (<i>ссылка на сообщение</i>).	
Результат	Пользователь успешно авторизирован и может работать с системой.
Расширения:	
*a	Нет доступа к БД. Система выдает сообщение (<i>ссылка на сообщение</i>). Результат: пользователь не может войти.
1a	В настройках безопасности для данного IP адреса существует запрет на вход в систему. Результат: форма логина не предоставляется, система выдает сообщение пользователю (<i>ссылка на сообщение</i>).
2a	Пользователь выбирает: «Напомнить пароль». Вызывается сценарий «Напомнить пароль».
3a	Пользователь с введенными логином и паролем не найден. Результат: отказ в авторизации. Система выдает сообщение (<i>ссылка на сообщение</i>). Переход на шаг 2.

Тест-анализ:

User Story (Пользовательская История)

- это краткое, точное и понятное описание функционала продукта или его особенностей, представляющих ценность для пользователя.

Заголовок	—	Краткое описание истории
ФОРМУЛИРОВКА ИСТОРИИ:		
Я как	—	Роль
Хочу	—	Функционал
Для того чтобы	—	Выгода

Как тестировать при отсутствии документации:

Алгоритм действий:

- задавать вопросы
- задавать вопросы
- задавать вопросы
- задавать вопросы
- задавать вопросы
- гуглить

Тестовая документация:

Характеристики требований:

1. Полнота. Можно вести чек-лист проверок или рисовать майнд-карту
2. Однозначность. Требования нужно конкретизировать. Не должно быть разночтений.
3. Непротиворечивость.
4. Ёмкость и краткость.
5. Осуществимость и реализуемость.
6. Тестируемость.

Техники тест-дизайна:

Здравый смысл :)

1. Анализ граничных значений
2. Классы эквивалентности
3. Предугадывание ошибки
4. Причина-Следствие
5. Таблица принятия решений
6. Диаграмма состояний и переходов
7. Попарное тестирование/Pairwise

Предугадывание ошибки:

--> Техника, при которой тестировщик использует свои знания системы, интуицию и способность к интерпретации спецификации на предмет того, чтобы "предугадать" при каких входных условиях система может выдать ошибку.

Предугадывание ошибок — это способ предотвращения ошибок, дефектов и отказов, основанный на знаниях тестировщика, включающих:

- Историю работы приложения в прошлом;
 - Наиболее вероятные типы дефектов, допускаемых при разработке;
 - Типы дефектов, которые были обнаружены в схожих приложениях
- Эта техника всегда идет в комбинации с какой-то другой (только пользуясь предугадыванием качественного тестирования не будет), а также с большим опытом тестировщика плюс отличное знание тестируемого продукта.

Классы эквивалентности:

Алгоритм действий:

- Определить классы эквивалентности
- Проанализировать классы
- Применить

Классы эквивалентности - это одно или несколько значений ввода, к которым программное обеспечение применяет одинаковую логику.

Цель: сократить количество тестов

-- это техника, при которой мы разделяем функционал на группы эквивалентных по своему влиянию на систему значений.



Классы эквивалентности:

Класс эквивалентности – это входные данные которые обрабатываются нашим приложением одинаково. Либо же обработка которых приводит к одному и тому же результату.

Эквивалентное разбиение

Интернет - магазин продажи алкогольной продукции
Ограничения по возрасту 18-100 лет

Возраст

Нажать

Позитивные тесты:

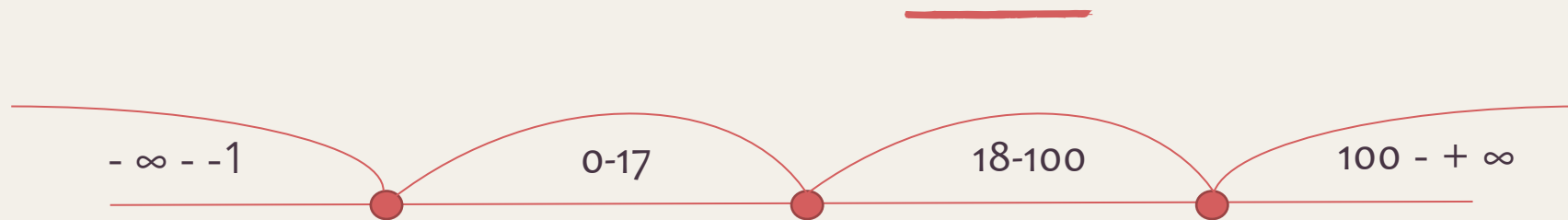
- 0-100 лет (**101** проверка)

Негативные тесты:

- $-\infty - -1 / 101 - + \infty$ (**бесконечное количество проверок**)
- Символы (**бесконечное количество проверок**)

Эквивалентное разбиение

Интернет - магазин продажи алкогольной продукции
Ограничения по возрасту 18-100 лет.



Классы эквивалентности

- 0-17 лет
- 18-100 лет
- $-\infty - -1$ лет
- $100 - + \infty$ лет
- Символы

Эквивалентное разбиение

Правила техники тест-дизайна эквивалентное разбиение:

- Определение классов эквивалентности
- Проведение 1 теста для 1 класса



Проверки для каждого класса эквивалентности

- 0-17 лет - 9 лет
- 18-100 лет - 41 год
- $-\infty - -1$ лет - -545
- $100 - +\infty$ лет - 545
- Символы - ты@

Анализ граничных значений:

Граничные значения могут быть у:

- Цифр
- Букв
- Месяцев/недель и т.д.

+ это может быть
сумма/произведение нескольких значений

--> Техника проверки поведения продукта на крайних (граничных) значениях входных данных.

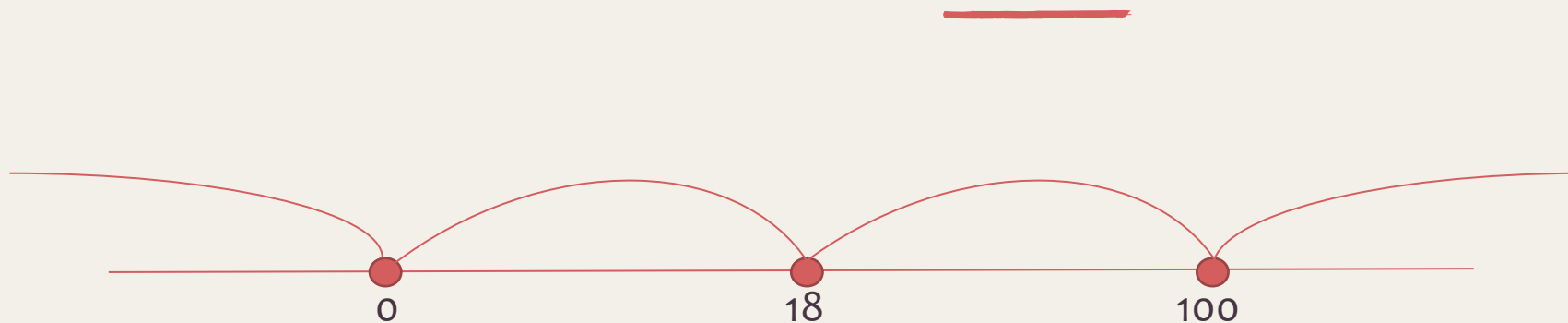
Цель: обнаружить ошибки и убедиться, что все работает верно на граничных значениях.

На каждой границе диапазона следует проверить по 3 значения:

- Граничное значение
- Значение перед границей
- Значение после границы

Анализ граничных значений

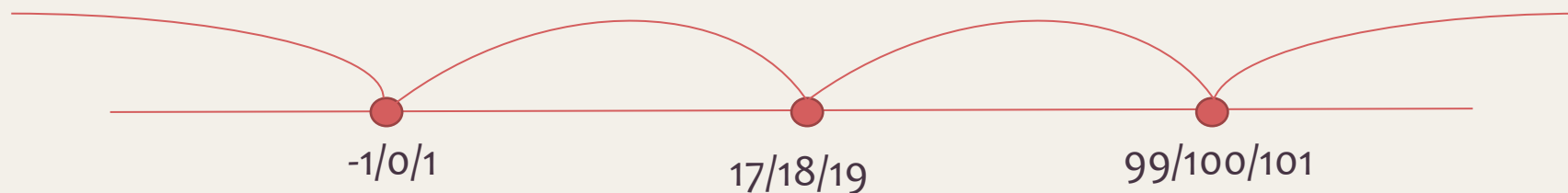
Граничные значения — это значения, в которых один класс эквивалентности переходит в другой диапазон.



Анализ граничных значений

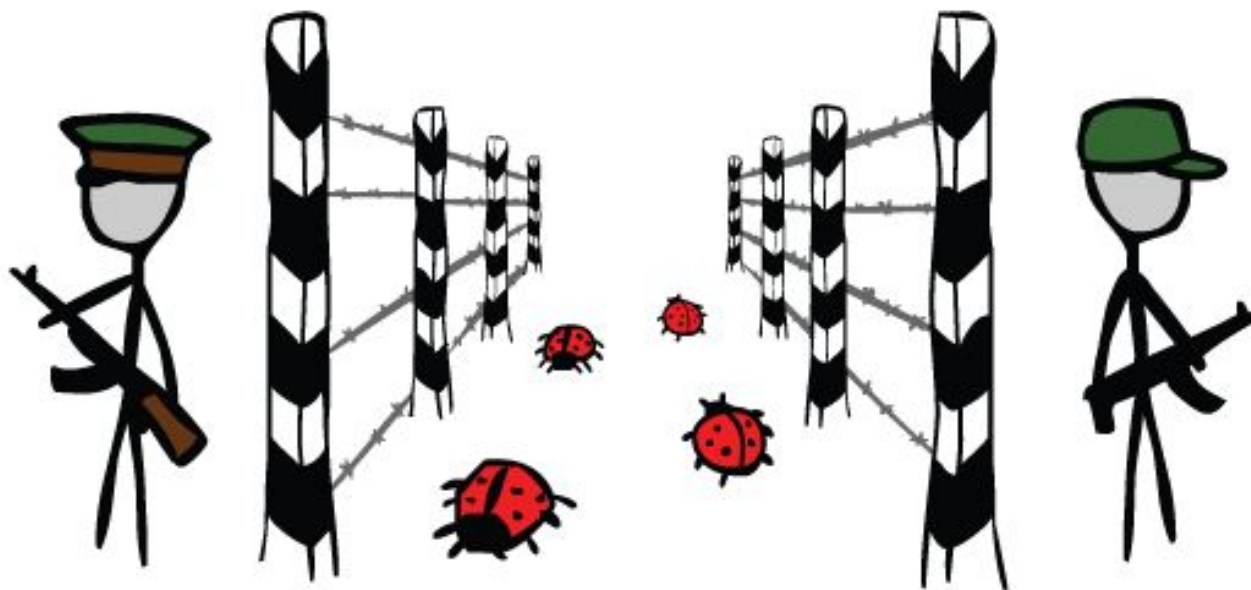
Правила техники тест-дизайна анализ граничных значений:

- Определение классов эквивалентности
- Определение границ диапазонов
- Проведение 3 тестов для границ: на самой границе , на +1 значении выше границы, на -1 значении ниже границы

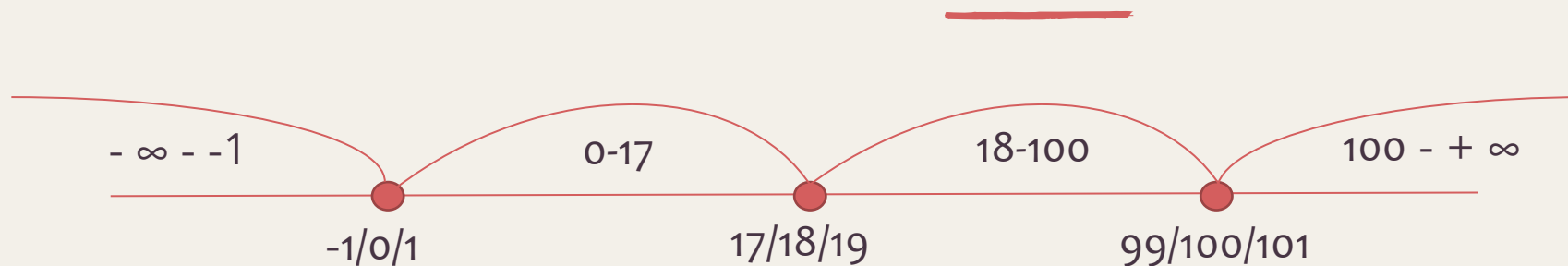


Анализ граничных значений

Баги водятся на границах



ИТОГ



Проверки для каждого класса эквивалентности

- 0-17 лет - 9 лет – 1 значение
- 18-100 лет - 41 год – 1 значение
- $-\infty - -1$ лет - -545 – 1 значение
- $100 - + \infty$ лет – 545 – 1 значение
- Символы - ты@ – 1 значение

Проверки для граничных значений

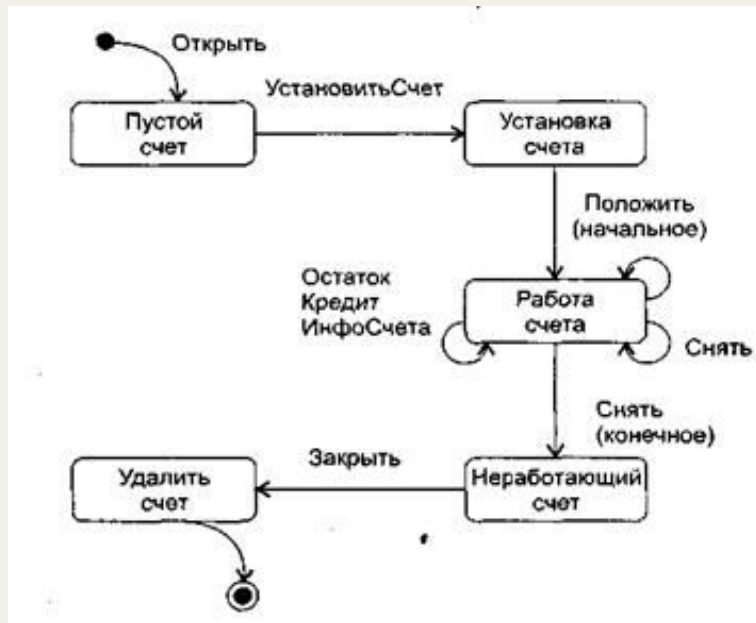
- $17/18/19$ – 3 значения
- $99/100/101$ – 3 значения
- $-1/0/1$ – 3 значения

Практическое задание:

Необходимо выбрать классы эквивалентности и граничные значения для тестирования формы возраста для покупки GTA 5



Диаграмма состояний и переходов:



Наглядно показывает, как некий объект переходит из одного состояния в другое. Есть события/ивенты (что делает пользователь), а в ответ от программы - действие/экшн.

Диаграмма состояний и переходов:

1. Переход может осуществлять как пользователь, так и система
2. Объект может находиться в одном состоянии
3. Объект должен быть один
4. Данный вид тестирования не про GUI



Таблица (матрица) принятия решений:

--> Способ компактного представления модели со сложной логикой, инструмент для упорядочения сложных бизнес-требований, которые должны быть реализованы в продукте.

Условия	Значения	Правила				
Логин	Верный, Неверный	Верный	Верный	Верный	Неверный	Неверный
Пароль	Верный, Неверный	Верный	Верный	Неверный	Неверный	Верный
Код	Верный, Неверный, Не пришел	Верный	Неверный	Не пришел	Не пришел	Не пришел
Действия	Пользователь вошел на сайт Пользователь не авторизован	Пользователь вошел на сайт	Пользователь не авторизован	Пользователь не авторизован	Пользователь не авторизован	Пользователь не авторизован

Попарное тестирование/Pairwise:

--> Суть этого метода в том, что каждое значение каждого проверяемого параметра должно быть протестировано на взаимодействие с каждым значением всех остальных параметров. После составления такой матрицы мы убираем тесты, которые дублируют друг друга, оставляя максимальное покрытие при минимальном необходимом наборе сценариев.

Браузер: Chrome, Opera

ОС: Windows, Linux

Язык: RU, ENG

Google Chrome	Windows	RU
Google Chrome	Windows	EN
Google Chrome	Linux	RU
Google Chrome	Linux	EN
Opera	Windows	RU
Opera	Windows	EN
Opera	Linux	RU
Opera	Linux	EN



Google Chrome	Windows	RU
Google Chrome	Linux	EN
Opera	Windows	EN
Opera	Linux	RU

<https://pairwise.teremokgames.com/>

Попарное тестирование/Pairwise:

--> метод создания комбинаций в котором с помощью пар можно найти максимальное количество дефектов.
Именно при использовании двух данных происходит наибольшее количество дефектов.
По статистике 97%

Причина-следствие:

Это ввод комбинаций условий (причин) для получения ответа от системы (следствий).

Эта техника помогает:


- Определить минимальное количество тестов для нахождения максимума ошибок.
- Выяснить все причины и следствия – таким образом, мы убедимся, что на любые манипуляции с системой у системы будет ответ.
- Найти возможные недочеты в логике описания приложения (что, в свою очередь, поможет улучшить документацию).

Типы полей ввода:

Текстовое поле

Введите имя

Поля для ввода даты и времени

05.12.1980 

Браузеры обычно отображают возле поля календарь, с помощью которого можно заполнить поле.

Существует несколько типов полей, содержащих данные о дате и времени:

- *date* - дата (число, месяц, год);
- *datetime-local* - дата и время (число, месяц, год, часы, минуты);
- *month* - месяц конкретного года (например: январь, 2004г);
- *week* - неделя конкретного года (например: неделя 32, 2001г).

Чек-бокс

Принимаю условия

Чек-бокс представляет собой поле, в котором можно сделать отметку.

Поле с указанным типом содержимого

Поле для ввода пароля:

.....

В поле типа *password* введенное значение будет заменено символьной маской.

Поле для ввода чисел:

1024

В поля из этой группы можно вводить только подходящее к типу поля содержимое.

Поля с указанным типом содержимого:

- *email* - поле для ввода адреса электронной почты;
- *number* - поле для ввода чисел;
- *password* - поле для ввода пароля;
- *search* - поле для ввода поискового запроса;
- *tel* - поле для ввода номера телефона;
- *url* - поле для ввода URL адреса.

Переключатель

- Самовывоз
- Курьер
- Экспресс-почта

Элемент "переключатель" позволяет выбрать один из доступных вариантов.

Типы полей ввода:

Текстовая область

Введите сообщение

Текстовая область может принимать в виде значения многострочный текст.

Выпадающий список

32 Гб, Черный ▾

При клике на список пользователю откроются доступные для выбора варианты.

Поле загрузки файлов

Выберите файл | Файл не выбран

С помощью этого элемента пользователь может прикрепить к форме файл.

Список с
множественным
выбором

Понедельник
Вторник
Среда
Четверг

Пользователь может выбрать сразу несколько вариантов из этого списка.

Кнопка

Отправить

Checkboxes

Option 1

Option 2

Option 3

Option 4

Radio Buttons

Option 1

Option 2

Option 3

Option 4

Проверки типов данных:

Файлы:

Размер, формат, имя, существование, права.

Дата-время:

Границы минут, часа, года, месяца и т.д.
Переходы зима/лето, несуществующие даты, часовые пояса, отношение к текущему моменту, взаимодействие дат.

Текстовое поле:

Буквы, цифры, спецсимволы, пробел (в начале, в конце, в середине, только пробел), alt-символы, редактирование строки, символы конца строки (/n, /r), инъекции, пустое поле.

Числовое поле:

Положительные, отрицательные, ноль, степени 2, дробные числа, формулы, числовые выражения, пустое поле.

Радиокнопки/чекбоксы:

Выбор одного варианта, всех, ни одного

Практическое задание:

Задание1:

Вам в тест пришла задача, в которой нужно протестировать отображение новогодней акции.

Сроки акции с 20 декабря по 10 января.

Напишите, как бы вы ее проверяли.

Задание2:

Составьте матрицу принятия решений для след.задачи.

Студент на сайте школы выбирает Курс для себя.

Курсы: Тестирование ПО, Python, SQL для начинающих, SQL для выпускников.

Даты начала: 20.12, 25.12, 10.01, 01.02, 15.02

Радиобаттон "Проходили ли уже курсы у нас?" Да/Нет.

Курс Тестирования начинается 25.12 или 10.01

Курс Python стартует 25.12, 10.01, 01.02

Курс SQL1 01.02

Курс SQL2 25.12 и 15.02

Проверка формы:

Тестирование веб-формы

<http://testingchallenges.thetestingmap.org/>

<https://playground.learnqa.ru/puzzle/triangle>

Домашка:

Задание 1 (normal)

Есть форма ввода даты рождения в графическом интерфейсе. Формат ввода ДД.ММ.ГГГГ. Допустимый период - [1900-2030гг]. Подсказок к заполнению поля нет, маски ввода даты нет.

Составьте таблицу, в которой будет 2 столбца: слева тестовое значение, справа результат и цель проверки. Чем больше проверок у вас получится - тем лучше! Ответ приложите файлом.

Задание 2 (normal)

Вы тестируете что-то вроде калькулятора. Есть 2 поля, между ними знак деления, под полями кнопка "Посчитать!". Напишите, пожалуйста, какие тестовые данные вы бы использовали для проверки.

Домашка:

Задание 3 (hard)

Вам необходимо протестировать программу для покупки/продажи бывшей в употреблении бытовой техники, есть следующие входные данные:

- категория заказа: покупка, продажа;
- местоположение: Москва, Санкт-Петербург;
- марка бытовой техники: Bosch, Indesit, Samsung;
- возможность работы с "умным домом": доступна, недоступна;
- тип расчета: наличный, безналичный;
- тип доставки: почтой, личная встреча.

Напишите, какую технику вы бы использовали для тестирования, почему выбрали именно эту технику и сами тестовые данные.