

Отчет о выполнении тестового задания

на направление «Тестирование»

Выполнил: Нам Андрей Владимирович

Томск 2022 г.

Задание 1

Жизненный цикл ПО начинается с того, что появляется идея продукта, а заканчивается полным выводом продукта из эксплуатации. Рассмотрим подробнее каждый этап и какое значение тестировщика в каждом из них.

Тестировщик подключается на этапе анализа продукта, когда подготавливаются требования, чем раньше тестировщик подключится к проекту тем меньше временных и денежных затрат потребуется на проект, так как чем позже обнаружен баг, тем дороже его потом устранить.

На этапе проектирования тестировщик может быть подключен к тестированию UI/UX. Для того чтобы проверить насколько интерфейс соответствует составленным ранее требованиям.

В течение этапа разработки тестировщик проводит модульное, интеграционное и системное тестирование. То есть сначала проверяются отдельные модули, потом несколько объединенных модулей и в конце тестируется система целиком.

Следующим этапом идет полное тестирование написанного функционала и тестирование интерфейса. На данном этапе применяется регрессионное тестирование. Это важнейший этап, поскольку продукт буквально в шаге от внедрения и от тестировщика зависит качество софта и репутация компании.

Тестировщик занимает немаловажную роль во всем жизненном цикле ПО и участвует на всех этапах разработки. Гиппократ завещал: «Болезнь легче предупредить, чем лечить».

Задание 2

В данном задании мы можем заметить, что проблема возникла на этапе авторизации пользователя. В данной ситуации дальнейшие действия могут быть следующими:

- 1) Связаться с пользователем и попросить предоставить видео попытки авторизации или попросить описать точные шаги и предоставить информацию. Чтобы уточнить корректность ввода данных.

Спросить с какого веб-браузера происходила авторизация.

- 2) Уточнить дату и время, когда пользователь производил авторизацию, для того чтобы уточнить не проводились ли какие-либо тех. работы, из-за которых могла бы возникнуть данная ошибка.

- 3) Спросить у разработчиков в чем может быть проблема. Так как на приложенном скриншоте, отчет об ошибке не сообщает пользователю о конкретной проблеме.

Также, не предложены определенные способы решения проблемы.

Не предоставлена информация о том куда пользователь может обратиться за помощью(техническая поддержка).

- 4) Завести баг-репорт с высоким приоритетом(так как данный дефект прекращает работу приложения), без авторизации в приложении дальнейшая работа в нем невозможна.

Задание 3

Ответ: Е

Решение:

Одно из основных условий это исключающее ИЛИ, а именно мы имеем что все автотесты выполнялись корректно (зеленые), либо красные. Тогда можно исключить варианты, что все автотесты красные и что все автотесты зеленые (так как они либо все красные, либо зеленые по условию). Также исключаем вариант что все автотесты НЕ зеленые, так как снова по условию они либо зеленые, либо красные. Вариант что все автотесты написаны тестировщиком Кеном тоже не подходит так как мы не знаем какого цвета автотесты написанные тестировщиком Джоном. Тогда ответ: Е.

Задание 4

Формула для вычисления следующая:

Где: i – ряд,

j – ячейка,

n – номер ряда.

Сообщение, выводимое клиенту:

Номер вашей ячейки – 35.

Задание 5

Если старшей дочери и матери в сумме 55 лет и у них одинаковые цифры, то имеется 2 варианта.

Рассмотрим вариант, когда старшей дочери 23 года, а матери 32.

Так как в возрасте младшей дочери те же цифры, которые будут в возрасте матери через количество лет, равное возрасту старшей дочери минус возраст младшей дочери сейчас, то можно составить следующее выражение:

Мать+старшая дочь-младшая дочь – возраст матери спустя определенное кол-во лет

Упростим, 55 - младшая дочь

В числе, которое получится будут такие же цифры, что и в возрасте младшей дочери.

То есть возраст младшей дочери примем за x

А возраст матери тогда y

Тогда

$$55 - 10x-y=10y+x$$

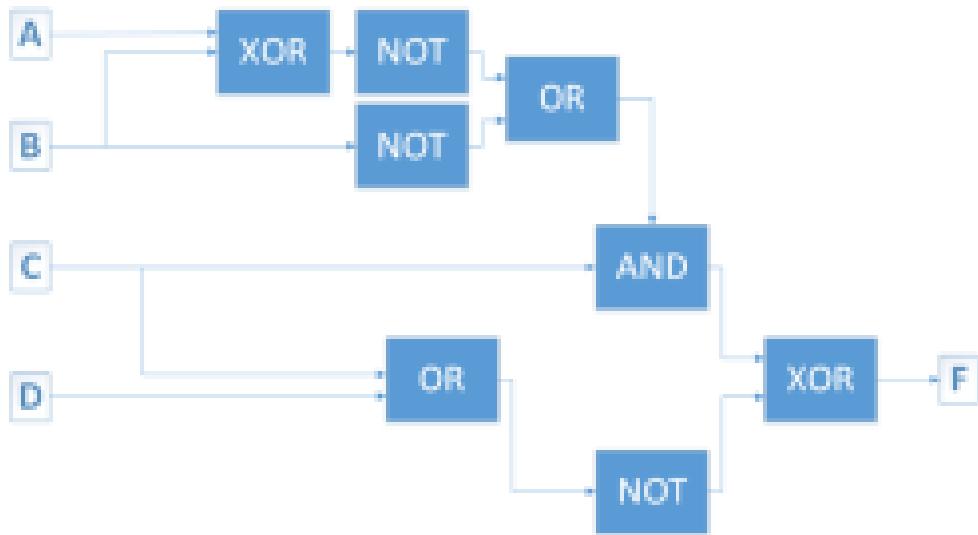
$$55=11(x+y)$$

$$x+y=5$$

Сумма цифр равна 5

Значит возраст младшей дочери будет равен 14

Задание 6



Необходимо найти значения A,B,C и D при которых F = истина(1).

Так как мы имеем 4 значения то вариантов их комбинаций $4^2 = 16$.
Но не все комбинации дадут истину.

Способ решения:

Начнем подбирать значения с конца.

Используем таблицу истинности:

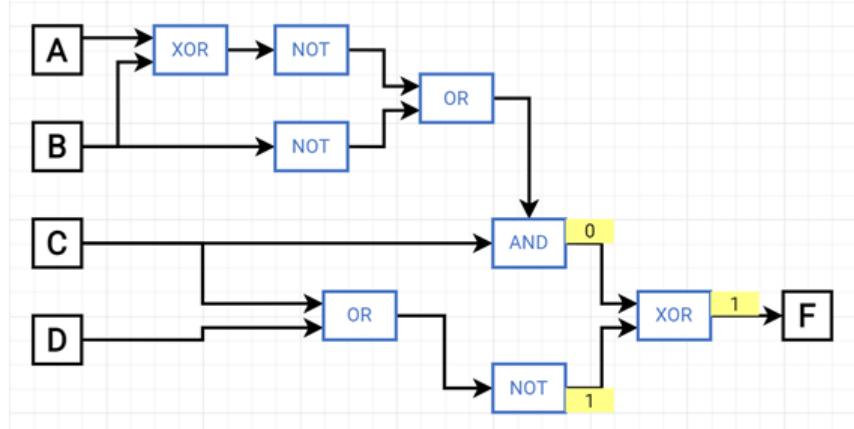
X	Y	NOT X	X AND Y	X OR Y	X XOR Y
0	0	1	0	0	0
0	1		0	1	1
1	0	0	0	1	1
1	1		1	1	0

Чтобы F было равно 1 необходимо чтобы операция XOR была равна 1.

Исключающее ИЛИ(XOR) равно истине(1) тогда, когда аргумент равен 1 а другой равен 0.

Таким образом входные значения в операцию XOR должны быть равны 0 и 1 или 1 и 0.

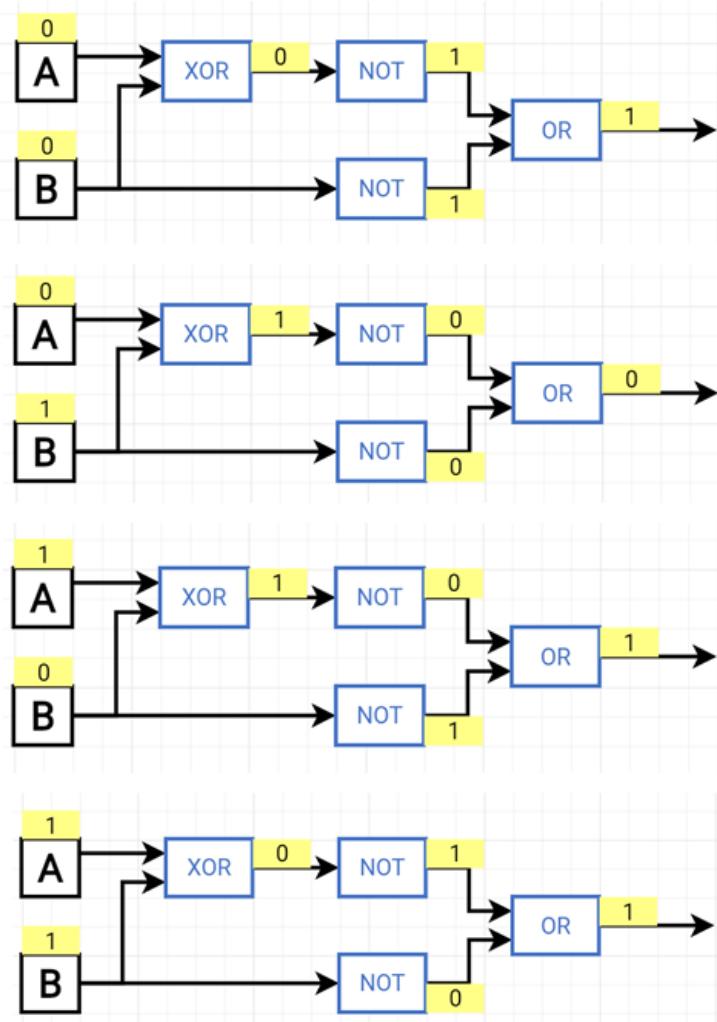
- 1) Рассмотрим вариант входных значений 0 XOR 1 = 1.



Чтобы операция AND принимала значение ложь(0) необходимо чтобы любое из входных значений равнялось 0. NOT станет истина, если изначально было значение ложь. OR будет равно 0, если значения C и D равны 0.

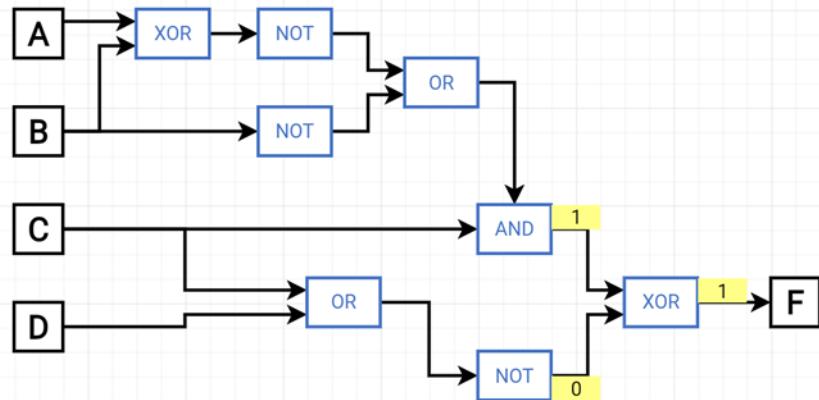
Соответственно: C=0, D=0.

Так как C = 0, то результат операций над значениями A и B может быть как истина, так и ложь. Рассмотрим все случаи входных значений A и B.



Итого при варианте входных значений 0 XOR 1 = 1, имеем 4 варианта значений (указаны по порядку ABCD): 0000, 0100, 1000, 1100.

2) Рассмотрим вариант входных значений $1 \text{ XOR } 0 = 1$.



Чтобы AND был равен 1, необходимо, чтобы оба входных значений было равно 1. Значит С=1. Значения входных данных А и В возьмем выше из таблиц, где результат операции $\text{OR} = 1$ ($A=0$ и $B=0$; $A=1$ и $B=0$; $A=1$ и $B=1$). Чтобы NOT получил значение 0, необходимо входное значение 1, а так как С имеет значение истины, то D может быть как 0, так и 1.

Таким образом получаем 6 вариантов значений (указаны по порядку ABCD): 0010, 0011, 1010, 1011, 1110, 1111.

В итоге получаем 10 вариантов значений входных переменных, чтобы результат операций был равен истине: 0000, 0100, 1000, 1100, 0010, 0011, 1010, 1011, 1110, 1111.

Задание 7

Тестирование программного обеспечения

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

В программировании баг (англ. bug — первичные значения: клоп, любое насекомое, вирус) — жаргонное слово, обычно обозначающее ошибку в программе или системе, из-за которой программа выдает неожиданный результат.

Этимология

По одной из версий, в отношении программной ошибки этот термин впервые применила в 1946 году Грейс Холлер, программирующая в Гарвардском университете вычислительную машину Harvard Mark II. Прославив ошибку в работе программы до электромеханического реле машины, она нашла мотылька, застрявшего между контактами. Извлечённое насекомое было включено скотчем в технический дневник с сопроводительной надписью: «Первый реальный случай обнаружения жучка» (англ. first actual case of bug being found).

Разновидности баг

Борбаг — легко обнаруживаемый стабильный баг

Гейзенбаг — сложно обнаруживаемый и периодически исчезающий баг при попытке его обнаружения

Мандельбаг — баг с очень сложным, хаотичным, поведением

Шредингербаг — критическая ошибка, которая не проявляется пока кто-нибудь на ней не напечатается в исходном коде, после чего программа совершенно перестает работать

пример

При тестировании очередного релиза получилась следующая статистика багов по критериям

- приоритет
- функциональность

Приоритет/Функциональность	Ф1	Ф2	Ф5	Ф4	Ф3
П1	1	0	8	1	3
П2	0	4	0	0	
П3	1	0	0	4	1
П4	0	1	2	1	

Тестирование — это не поиск ошибок! Тестирование — это забота о качестве продукта в виде обнаружения багов до того, как их найдут пользователи.

Тестирование — это не поиск ошибок! Тестирование — это забота о качестве продукта в виде обнаружения багов до того, как их найдут пользователи.

Задание 8

Тест-кейс 1

Название:	Проверить подключения автомата к питанию	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автоматау		
Шаги теста:		
Проверить кабель питания	Кабель питания подключен	

Тест-кейс 2

Название :	Проверить выдачу 1 шарика шоколадного мороженого	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автоматау		
Шаги теста:		
Внести 40 рублей	Внесена оплата	

Выбрать 1 шарик шоколадного мороженого	Шарик выбран	
Нажать на кнопку выдачи	Выдан 1 шарик шоколадного мороженого	

Тест-кейс 3

Название :	Проверить выдачу 1 шарика клубничного мороженого	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести 30 рублей	Внесена оплата	
Выбрать 1 шарик клубничного мороженого	Шарик выбран	
Нажать на кнопку выдачи	Выдан 1 шарик клубничного мороженого	

Тест-кейс 4

Название :	Проверить выдачу 1 шарика ванильного мороженого	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести 50 рублей	Внесена оплата	
Выбрать 1 шарик ванильного мороженого	Шарик выбран	
Нажать на кнопку выдачи	Выдан 1 шарик ванильного мороженого	

Тест-кейс 5

Название :	Проверить выдачу топпинга посыпки	
Действие	Ожидаемый результат	Результат теста:

Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести 8 рублей	Внесена оплата	
Выбрать топпинг посыпка	Топпинг выбран	
Нажать на кнопку выдачи	Выдано 10 мл. топпинг посыпка	

Тест-кейс 6

Название :	Проверить выдачу топпинга сироп	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести 15 рублей	Внесена оплата	
Выбрать топпинг сироп	Топпинг выбран	

Нажать на кнопку выдачи	Выдано 10 мл топпинг сироп	
-------------------------	----------------------------	--

Тест-кейс 7

Название :	Проверить возможность покупки нескольких шариков мороженого	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги	Внесена оплата	
Выбрать несколько шариков	Шарики выбраны	

Тест-кейс 8

Название :	Проверить возможность получения бесплатного топпинга при покупке 3 шариков мороженого	
Действие	Ожидаемый результат	Результат теста:

Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги	Внесена оплата	
Выбрать покупки 3 шарика мороженого	Выбрано 3 шарика мороженого	
Выбрать топпинг	Топпинг выбран Топпинг бесплатный	
Нажать на кнопку выдачи	Мороженое с топпингом выдано	

Тест-кейс 9

Название :	Проверить отсутствие возможности покупки более 3 шариков мороженого	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		

Шаги теста:		
Внести деньги	Внесена оплата	
Выбрать покупки более 3 шариков мороженого	Выбрано максимум 3 шарика мороженого	

Тест-кейс 10

Название :	Проверить возможности внесения оплаты (1,2,5,10,100,500 рублей.)	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги (1,2,5,10,100,500 рублей.)	Внесена оплата	

Тест-кейс 11

Название:	Проверить доступных способов получения сдачи (1,2,5,10,100 рублей)	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги (1,2,5,10,100,500 рублей.)	Внесена оплата	
Выбрать мороженое	Мороженое выбрано	
Нажать на кнопку выдачи	Мороженое выдано	
Получить сдачу (1,2,5,10,100 рублей)	Сдача выдана (1,2,5,10,100 рублей)	

Тест-кейс 12

Название:	Проверить отсутствие возможности внесения оплаты непредусмотренными номиналами
------------------	--

Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги любые кроме (1,2,5,10,100,500 рублей.)	Купюры/монеты не приняты	

Тест-кейс 13

Название:	Проверить отсутствие возможности совершить несколько покупок	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги	Внесена оплата	

Совершить первый выбор	Выбор совершен	
Совершить второй выбор	Выбор запрещен	

Тест-кейс 14

Название:	Проверить возможность совершить одну покупку при одной оплате	
Действие	Ожидаемый результат	Результат теста:
Предусловие:		
Подойти к автомату		
Шаги теста:		
Внести деньги	Внесена оплата	
Совершить первый выбор	Выбор совершен	

Вопросы к требованиям автомата:

- 1)Удобный и понятный интерфейс для пользователей разного возраста.
- 2)Надежность(минимум 4 часа беспрерывного использования).
- 3)Компактность(аппарат не должен занимать больше отведенного ему места).
- 4)Наличие запасного аккумулятора(на случай перепадов в напряжении).
- 5)Наличие заземление у аппарата(безопасность пользователей).
- 6)Влагостойкость(на случай если аппарат находится не в помещении).
- 7)Удобность в транспортировке(у аппарата в его основании должны быть маленькие колеса)
- 8)Встроенный GPS трекер(на случай нежелательной кражи аппарата)