

Тестирование.
Идеи автоматизации тестирования.

Факты о тестировании

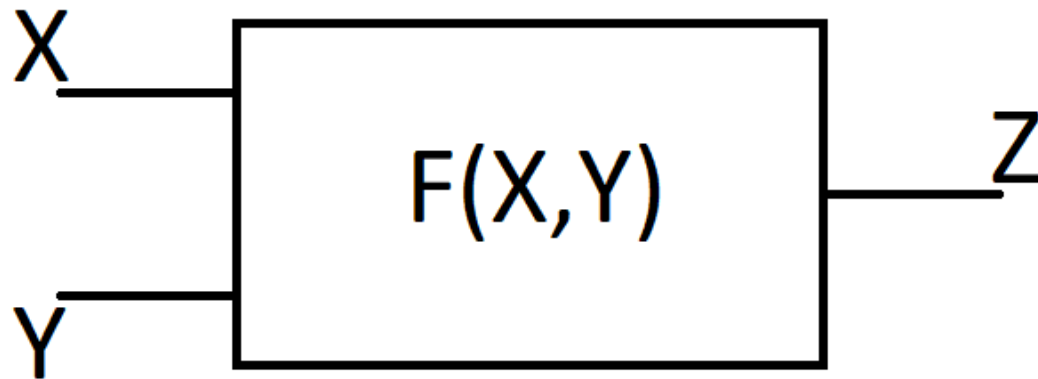
- «Тестирование может показать лишь наличие ошибок, но не их отсутствие.» Дейкстра Э.В.
- Невозможно исчерпывающее тестирование программы.

Тестирование показывает наличие ошибок

Сколько бы белых лебедей мы ни видели, это еще не служит основанием утверждать, что «все лебеди белые». Но если мы увидим хотя бы одного черного лебедя, мы можем сказать, что «не все лебеди белые». [Слинявчук В., luxoft]



Невозможность исчерпывающего тестирования



- Пусть X и Y могут принять 2^{32} значений (т.е. 2^{64} сочетаний).
- Программа работает одну миллисекунду.
- Чтобы протестировать все сочетания X и Y , программу пришлось бы тестировать в течение примерно 300 миллионов лет.

Как можно определить, что программа испытана достаточно полно?

Нужно продолжать тестирование до тех пор, пока каждая команда не будет использована хотя бы один раз.

Как можно определить, что программа испытана достаточно полно?

```
int apply_discount(int total)
{
    int discount;

    if (total >= 1000)
        discount = total * 0.05;

    return total - discount;
}

printf("%d\n", apply_discount(1025)) ;
```

Как можно определить, что программа испытана достаточно полно?

100% покрытие кода не гарантирует полную протестируемость программы/функции.

Классы тестовых данных

Мы не располагаем средствами, чтобы опробовать программу при всех возможных совокупностях данных, поэтому желательно подобрать наиболее характерные данные.

Группу значений, которые программа обрабатывает одним и тем же способом, назовем *классом эквивалентности*.

В рамках одного класса тесты приводят к одному и тому же результату, т.е. являются эквивалентными.

Классы тестовых данных

Как только удастся выделить класс, т.е. группу однотипных тестов, можно провести несколько из них и проигнорировать остальные.

Для отбора проводимых тестов есть важное правило: «для выполнения всегда выбирайте из класса те тесты, на которых вероятнее всего ожидается сбой программы».

Классы тестовых данных

Граничными значениями класса являются те данные, на которых программа меняет свое поведение.

При программировании граничных условий случайная ошибка очень вероятна, поэтому граничные данные следует проверять наиболее тщательно.

Пример (постановка задачи)

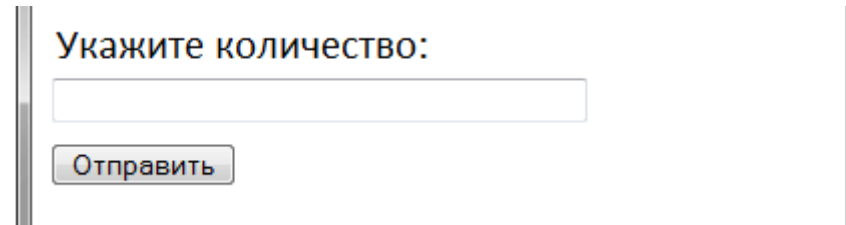
Компания занимается оптовой продажей некоторой продукции.

Компания предоставляет скидку в зависимости от количества покупаемых единиц продукции:

Единицы (штуки)	Скидка (%)
10 – 99	0
100 – 199	10
200 – 1000	20

Пример (постановка задачи)

У компании есть «on-line магазин». На одной из страниц этого магазина расположено поле ввода, в которое вводится количество закупаемых единиц товара. В поле ввода можно указывать любые «печатные» символы.



Укажите количество:

Отправить

Пример (классы эквивалентности)

...	10	100	200	1000	...
-----	----	-----	-----	------	-----

Пример (классы эквивалентности)

...	0	9	10	100	200	1000	...
K1	K2	K3	K4	K5	K6		

Пример (классы эквивалентности)

- K1 – до 0
- K2 – от 0 до 9
- K3 – от 10 до 99
- K4 – от 100 до 199
- K5 – от 200 до 1000
- K6 – свыше 1000

Пример (контрольные примеры)

- K1: -10 (сообщение о некорректности данных)
- K2: 5 (сообщение о том, что партия слишком маленькая)
- K3: 15
- K4: 132
- K5: 578
- K6: 3424 (сообщение о том, что партия слишком большая)

Плюс взять значения на границах каждого класса: -1, 0, 1, 9, 10, 11, 99, 100, 101, 199, 200, 201, 999, 1000, 1001.

Пример (проверка правильности числа)

- Правильное целое число
- **Не целое число**
- Пустая строка

Пример (проверка правильности числа)

- Правильное целое число
- **Не целое число**
 - только буквы
 - правильное действительное число
 - буквы и цифры
 - и т.д.
- Пустая строка

Задача

Выделить классы эквивалентности для программы, которая решает квадратные уравнения (ограничимся только действительными корнями).

Задача (ответ)

Тест	Проверяемый случай	Коэффициенты	Результат
1	$D > 0$	1 1 -2	Два корня: 1 -2
2	$D == 0$	1 2 1	Один корень: -1
3	$D < 0$	2 1 2	Действительных корней нет
4	$a == 0, b == 0, c == 0$	0 0 0	Бесконечно много решений
5	$a == 0, b == 0, c != 0$	0 0 2	Неправильное уравнение
6	$a == 0, b != 0, c != 0$	0 2 1	Линейное уравнение: -0.5
7	$a != 0, b != 0 \parallel b == 0, c == 0$	2 1 0	Неполное квадратное уравнение: 0 -0.5

Функциональное тестирование

Функциональное тестирование - это тестирование ПО в целях проверки реализуемости функциональных требований, то есть способности ПО в определённых условиях решать задачи, нужные пользователям. [Wikipedia]

Функциональное тестирование: идеи автоматизации

- Тестовые данные хранятся в файлах.
- Перенаправление ввода/вывода для использование файлов.
- Скрипты для автоматизации.