ЛАБОРАТОРНАЯ РАБОТА №4

Тема лабораторной работы: методы тест-дизайна.

Общие сведения по работе

Методы тест-дизайна упорядочивают деятельность по проектированию и созданию необходимых наборов тестов для контроля за качеством реализации программных продуктов. Основная задача при этом – минимизация количества тестов и количества их прогонов при максимизации контроля в условиях ограниченных временных, вычислительных и человеческих ресурсов. Частично требования к тест-дизайну изложены в требованиях к тестовой документации, стандартах процессов тестирования, требованиях к использованию инструментов тестирования, а также в описаниях этапов жизненного цикла разработки выбранной модели. В данной лабораторной работе необходимо сосредоточиться на задаче проектирования необходимого набора тест-кейсов для тестирования отдельной функциональности приложений, основываясь на анализе входной информации.

Методические рекомендации и материалы

Одной из проблем при создании набора тестов является разнообразие входных данных. Данные могут подавать на вход программной системы в разном сочетании, в разном объеме. Поэтому проектирование тестов должно учитывать и такие особенности.

Для минимизации выполняемых прогонов тестов применяют техники анализа эквивалентных классов, граничных значений. Сутью подходов является разделение всех подаваемых на вход данных на отдельные группы. Вся совокупность данных в пределах одной группы приводит к одной и той же реакции системы, в то время как данные из разных групп позволяют получить разный результат. Так, например, если тестируются

поля ввода текста и установлено ограничение на минимально допустимую длину текста, равное 5 символам, то ввод строк длиной 1,2,3 или 4 символов должен приводить к одной реакции системы – выводу ошибки. Таким образом, эквивалентный класс – это набор данных или тестов, проверяющих один из вариантов реализации логики функционирования программной системы. Минимизация тестов, выполняемых над системой, достигается за счет использования только одного из значений из эквивалентного класса.

Другая проблема, которая требует рассмотрения – это тестирование поведения системы в граничных значениях эквивалентных классов. Обосновывается эта проблема необходимостью контролировать покрытие всех возможных вариантов условными операторами (например, проверятся код: if (a.equals(“some_string”) в случаях a = null, а не содержит строку “some_string”)), вариантов использования операторов сравнения (x > y или x ≥ y ?) в программном коде. Способом проконтролировать поведение программной системы в таких случаях является анализ эквивалентных классов и граничных между ними значений. Методика тестирования будет заключаться в определении отношения граничных значений к определенному эквивалентному классу. После этого производится выполнение тестов на значениях каждого класса, плюс выполняются прогоны тестов на граничных значениях и в некоторой близкой области к граничным значениям. Например, необходимо протестировать ввод целочисленного положительного числа при условии, что поведение алгоритма меняется при значении переменной, равном 7. Эквивалентные классы будут включать два интервала: [0, …, 7] и (7, …, ∞); При использовании методики тестирования граничных значений нужно выполнить тесты при значении переменной, равном:

– значению из интервала [0, …, 7], например 3,

– значению переменной, равном значению на границе классов, т. е. 7,

– значениям переменной в области близкой к граничному значению (т.к. число целое, то минимальный сдвиг от границы может дать значения, равные 6 и 8),

– значению из интервала (7, …, ∞), например 9.

Методика тестирования предполагает использование всех перечисленных случаев, хотя очевидно, что значения 8 и 9 принадлежат одному классу и могут быть сведены только к одному значению. Использовать или не использовать все указанные значения решают на основе данных: если сдвиг от граничных значений и выбор дополнительных значений эквивалентных классов помогут повысить вероятность локализации дефектов, то нужно использовать все изложенные варианты данных. В случае целочисленных данных, как в примере, сложно сказать, что поведение системы может измениться, поэтому список данных, используемых в тестировании, можно сократить.

Задания к лабораторной работе

1. Для одной из форм приложения выделить эквивалентные классы.
2. Сделать расчет количества тестов для проверки формы приложения с учетом требования минимизации количества проводимых тестов.
3. В отчет по лабораторной работе включить:
    a. Цель работы.
    b. Список используемых тест-кейсов.
    c. Описание эквивалентных классов.
    d. Расчет количества тестов.
    e. Выводы по работе.
    f. Список использованных источников.
4. Оформить и защитить отчет.

Контрольные вопросы

1. Опишите методику выделения эквивалентных классов.
2. В чем цель тестирования граничных значений?
3. Что такое методика черного ящика?
4. В чем разница между методикой черного, белого и серого ящиков?
5. Что представляет собой тест-дизайн?