**Практическое занятие № 46**

**«Функциональное тестирование»**

**Цель:** получение навыков проведения функционального тестирования.

**Теоретические вопросы**

* особенности функционального тестирования программного обеспечения (тестирования «черного ящика»);
* ошибки, выявляемые при функциональном тестировании;
* задачи, решаемые при функциональном тестировании.

**Ответы на теоретические вопросы**

1. Особенности функционального тестирования ("черного ящика"):

* не требует знания внутренней структуры: тестировщик не знает внутренние детали реализации программы и тестирует ее только на основе ее внешних характеристик;
* основывается на требованиях: тесты разрабатываются на основе функциональных требований к программе;
* ориентирован на пользовательский опыт: целью функционального тестирования является проверка соответствия программы ожиданиям и потребностям пользователей;
* позволяет выявить разнообразные дефекты: тестирование может выявить ошибки в интерфейсе, логике приложения, обработке данных и других аспектах функциональности программы.

1. Ошибки, выявляемые при функциональном тестировании:

* неправильное поведение приложения: например, программа может выдавать неверные результаты или неправильно обрабатывать входные данные;
* проблемы с интерфейсом пользователя: некорректное отображение элементов интерфейса, неверные маршруты навигации, неправильное взаимодействие с пользователем;
* нарушение функциональных требований: например, программа может не соответствовать спецификации, установленным стандартам или ожиданиям заказчика;
* проблемы с производительностью: например, программа может работать медленно или терять данные из-за неправильной оптимизации;
* совместимость и безопасность: тестирование также может выявить проблемы совместимости с другими программами или платформами, а также уязвимости безопасности.

1. Задачи, решаемые при функциональном тестировании:

* проверка функциональности: убеждается, что все функции программы работают правильно и соответствуют ожиданиям пользователей;
* подтверждение соответствия требованиям: проверка того, что программа соответствует функциональным требованиям, установленным в спецификации или документации;
* поиск дефектов: выявление и документирование ошибок и несоответствий в поведении программы;
* улучшение качества программы: предоставление обратной связи разработчикам для улучшения качества программы и исправления обнаруженных проблем;
* повышение удовлетворенности пользователя: гарантия того, что программа соответствует потребностям и ожиданиям конечных пользователей.

Задание № 1

Опишите методы формирования тестовых наборов при использовании стратегии «черного ящика»:

Таблица 1 – Методы формирования тестовых наборов при использовании стратегии «черного ящика»

|  |  |
| --- | --- |
| Эквивалентное разбиение | Этот метод предполагает разделение входных данных на эквивалентные классы, в пределах которых ожидается, что программа будет вести себя одинаково.  На основе этого разбиения создаются тестовые случаи, чтобы обеспечить покрытие каждого класса эквивалентности.  Например, если программа должна обрабатывать числа от 1 до 100, можно выбрать по одному представителю из каждого класса, например, числа 0, 50 и 101, чтобы проверить, как программа обрабатывает недопустимые, типичные и крайние значения. |
| Анализ граничных значений | Этот метод сосредотачивается на проверке поведения программы в окрестности граничных значений входных данных.  Тестовые случаи создаются для проверки программы на граничных значениях и на значениях, лежащих непосредственно за ними.  Это помогает выявить ошибки, связанные с обработкой граничных условий.  Например, если программа принимает значения от 1 до 10, тестовые случаи могут быть созданы для проверки поведения при вводе 1, 10 и 11. |
| Анализ причинно-следственных связей | Этот метод используется для моделирования возможных комбинаций входных данных и их влияния на поведение программы.  Создается граф, который показывает взаимосвязи между входными и выходными данными программы.  На основе этого графа определяются тестовые случаи, которые покрывают различные комбинации входных данных и событий. |

Продолжение таблицы 1

|  |  |
| --- | --- |
|  | Этот метод может помочь выявить сложные взаимосвязи и проблемы в программе, которые могли бы быть пропущены при других методах тестирования. |
| Предположение об ошибках | Этот метод основан на опыте и интуиции тестировщика.  Тестировщик предполагает, где могут быть ошибки в программе на основе своего опыта, знаний о предметной области и характеристик программируемого приложения.  Тестовые случаи создаются на основе этих предположений с целью проверить программу на наличие этих ошибок. Хотя этот метод не всегда структурирован, он может быть полезным в выявлении ошибок, особенно там, где другие методы тестирования могут быть недостаточны. |

Задание № 2

Пусть необходимо выполнить тестирование программы, определяющей точку пересечения двух прямых на плоскости. Попутно, она должна определять параллельность прямой одной их осей координат.

В основе программы лежит решение системы линейных уравнений Ax + By = C и Dx + Ey= F.

1. Используя метод эквивалентных разбиений, получаем для всех коэффициентов один правильный класс эквивалентности (коэффициент – вещественное число) и один неправильный (коэффициент – не вещественное число). Откуда можно предложить 7 тестов:
   1. все коэффициенты – вещественные числа;
   2. поочередно каждый из коэффициентов – не вещественное число.

Таблица 2 – Тесты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Все коэффициенты – вещественные числа | A=3, B=7, C=4, D=2, E=5, F=9 | Программа определит точку пересечения прямых и выведет ее координаты. | Положительная | Точка пересечения определена, координаты верные. |
| 2 | A – не вещественное число | A=6i, B=5, C=10, D=2, E=7, F=3 | Программа выведет сообщение о некорректных данных и | Отрицательная | Программа вывела сообщение о некорректных данных и |

Продолжение таблицы 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | попросит ввести вещественные коэффициенты. |  | запросила введение вещественных коэффициентов. |
| 3 | B – не вещественное число | A=2, B=3i, C=11, D=6, E=7, F=1 | Программа выведет сообщение о некорректных данных и попросит ввести вещественные коэффициенты. | Отрицательная | Программа вывела сообщение о некорректных данных и запросила введение вещественных коэффициентов. |
| 4 | C – не вещественное число | A=4, B=3, C=5i, D=2, E=12, F=7 | Программа выведет сообщение о некорректных данных и попросит ввести вещественные коэффициенты. | Отрицательная | Программа вывела сообщение о некорректных данных и запросила введение вещественных коэффициентов. |
| 5 | D – не вещественное число | A=2, B=8, C=5, D=2i, E=3, F=1 | Программа выведет сообщение о некорректных данных и попросит ввести вещественные коэффициенты. | Отрицательная | Программа вывела сообщение о некорректных данных и запросила введение вещественных коэффициентов. |
| 6 | F – не вещественное число | A=8, B=6, C=3, D=4, E=7, F=9i | Программа выведет сообщение о некорректных данных и попросит ввести вещественные коэффициенты. | Отрицательная | Программа вывела сообщение о некорректных данных и запросила введение вещественных коэффициентов. |

Продолжение таблицы 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | B и D – не вещественные числа | A=6, B=9i, C=7, D=11i, E=8, F=2 | Программа выведет сообщение о некорректных данных и попросит ввести вещественные коэффициенты. | Отрицательная | Программа вывела сообщение о некорректных данных и запросила введение вещественных коэффициентов. |

1. По методу граничных условий можно считать, что для исходных данных граничные условия отсутствуют (коэффициенты – «любые» вещественные числа); для результатов – получаем, что возможны варианты: единственное решение, прямые сливаются (множество решений), прямые параллельны (отсутствие решений).

Следовательно, можно предложить тесты, с результатами внутри области и с результатами на границе.

Таблица 3 – Тесты, с результатами внутри области и с результатами на границе.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Прямые параллельны оси Ox | A=7, B=0, C=2, D=5, E=0, F=4 | Прямые параллельны оси Ox. | Положительная | Программа верно определяет, что прямые параллельны оси Ox |
| 2 | Прямые параллельны оси Oy | A=0, B=7, C=5, D=0, E=1, F=8 | Прямые параллельны оси Oy | Положительная | Программа верно определяет, что прямые параллельны оси Oy |
| 3 | Прямые не параллельны | A=3, B=4, C=11, D=5, E=2, F=7 | Прямые пересекаются в одной точке | Положительная | Программа верно определяет, что прямые пересекаются в одной точке |

1. По методу анализа причинно-следственных связей определяем множество условий:

а) для определения типа прямой; 3 на тип

б) для определения точки пересечения. 2 на точку пересечения

Выделяем три группы причинно-следственных связей (определение типа и существования первой линии, определение типа и существования второй линии, определение точки пересечения) и строим таблицы истинности.

Таблица 4 – Тесты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Наклонная прямая | A=1, B=1, C=5, D=2, E=1, F=2 | Прямые пересекаются | Положительная | Тест пройден |
| 2 | Горизонтальная прямая | A=2, B=0, C=6, D=1, E=1, F=4 | Прямые параллельны оси Y | Положительная | Тест пройден |
| 3 | Вертикальная прямая | A=0, B=3, C=9, D=0, E=2, F=5 | Прямые параллельны оси X | Положительная | Тест пройден |
| 4 | Прямые пересекаются | A=1, B=1, C=5, D=2, E=1, F=5 | Точка пересечения (0, 5) | Положительная | Тест пройден |
| 5 | Прямые параллельны | A=2, B=3, C=4, D=4, E=6, F=8 | Прямые параллельны, нет точки пересечения. | Положительная | Тест пройден |

К уже имеющимся тестам добавляются:

а) проверки всех случаев расположения обеих прямых – 6 тестов по первой прямой вкладываются в 6 тестов по второй прямой так, чтобы варианты не совпадали, – 6 тестов;

б) выполняется отдельная проверка несовпадения условия x = 0 или y = 0 (в зависимости от того, какой тест был выбран по методу граничных условий) – тест также можно совместить с предыдущими 6 тестами;

Таблица 5 – Продолжение тестов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Параллельные прямые | A=3, B=4, C=7, D=3, E=4, F=10 | Прямые параллельны | Положительная | Тест пройден |
| 2 | Пересекающиеся прямые | A=2, B=5, C=8, D=7, E=3, F=14 | Точка пересечения | Положительная | Тест пройден |
| 3 | Пересекающиеся прямые | A=3, B=1, C=4, D=1, E=3, F=2 | Точка пересечения | Положительная | Тест пройден |

Продолжение таблицы 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | Совпадающие прямые | A=1, B=2, C=3, D=1, E=2, F=3 | Бесконечное множество точек пересечения | Положительная | Тест пройден |
| 5 | Вертикальная прямая | A=0, B=1, C=1, D=1, E=0, F=3 | Прямые пересекаются | Положительная | Тест пройден |
| 6 | Горизонтальная прямая | A=1, B=0, C=1, D=2, E=0, F=1 | Прямые пересекаются | Положительная | Тест пройден |
| 7 | x ≠ 0 | A = 1, B = -1, C = 5, D = 1, E = 1, F = 3 | (2, 3) | Отрицательная | Точка пересечения не лежит на оси X |
| 8 | y ≠ 0 | A = -2, B = 3, C = 5, D = 1, E = 1, F = 3 | (2, 3) | Отрицательная | Точка пересечения не лежит на оси Y |

1. По методу предположения об ошибке добавим тест, при котором все коэффициенты – нули. Всего получили 20 тестов по всем четырем методикам. Если еще попробовать вложить независимые проверки, то возможно число тестов можно еще сократить.

Таблица 6 – Тест со всеми коэффициентами равными 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Все коэффициенты равны 0 | A=0, B=0, C=0, D=0, E=0, F=0 | Прямые некорректны | Отрицательная | Ошибка |

Задание № 3

Разработать программу определения вида треугольника, заданного длинами его сторон: равносторонний, равнобедренный, прямоугольный, разносторонний.

Таблица 7 – Предлагаемые тесты

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Равносторонний треугольник | a=3, b=3, c=3 | Треугольник равносторонний | Показывает сообщение "Треугольник равносторонний" | Тест пройден |

Продолжение таблицы 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2 | Равнобедренный треугольник | a=2, b=2, c=5 | Треугольник равнобедренный | Показывает сообщение "Треугольник равнобедренный" | Тест пройден |
| 3 | Прямоугольный треугольник | a=6, b=8, c=10 | Треугольник прямоугольный | Показывает сообщение "Треугольник прямоугольный" | Тест пройден |
| 4 | Разносторонний треугольник | a=7, b=4, c=9 | Треугольник разносторонний | Показывает сообщение "Треугольник разносторонний" | Тест пройден |
| 5 | Несуществующий треугольник | a=2, b=4, c=6 | Треугольника не существует | Показывает сообщение "Треугольник не существует" | Тест пройден |

Задание № 4

Разработать программу решения уравнения ax2 + bx +c = 0, где a, b, c – любые вещественные числа.

Таблица 8 – Предлагаемые тесты.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Назначение теста | Значения исходных данных | Ожидаемый результат | Реакция программы | Вывод |
| 1 | Проверка ввода некорректных значений | "qwerty", "44a/77d" | "Ошибка ввода. Пожалуйста введите числовые данные" | Положительная | MessageBox с сообщением о некорректных значениях |
| 2 | a = 0 | a=0, b=4, c=6 | Корни квадратного уравнения равен x1 = не число; х2 = -∞ | Положительная | MessageBox с корнями уравнения |
| 3 | b = 0 | a=1, b=0, c=7 | «У уравнения нет действительных корней» | Положительная | MessageBox об отсутствии корней |
| 4 | c = 0 | a=3, b=5, c=0 | Корни квадратного уравнения равны x1 = 0, x2 = -1.7 | Положительная | MessageBox с корнями уравнения |
| 5 | Дискриминант > 0 | a=3, b=6, c=1 | Корни квадратного уравнения равны x1 = -0,2, x2 = -1,8 | Положительная | MessageBox с корнями уравнения |

Продолжение таблицы 8

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 6 | Дискриминант = 0 | a=8, b=8, c=2 | Корень квадратного уравнения равен x = -0,5 | Положительная | MessageBox с корнем уравнения |
| 7 | Дискриминант < 0 | a=8, b=2, c=8 | «У уравнения нет действительных корней» | Положительная | MessageBox об отсутствии корней |