

Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных

Модульное тестирование — это проверка отдельных изолированных компонентов программы: функций, методов, классов или модулей. Его цель — выявить локализованные в модуле ошибки в реализации алгоритмов, а также определить степень готовности системы к переходу на следующий уровень разработки и тестирования.

Особенности модульного тестирования:

- проводится по принципу «белого ящика», то есть основывается на знании внутренней структуры программы, и часто включает методы анализа покрытия кода;
- обычно подразумевает создание вокруг каждого модуля определённой среды, включающей заглушки для всех интерфейсов тестируемого модуля;
- на уровне модульного тестирования проще всего обнаружить дефекты, связанные с алгоритмическими ошибками и ошибками кодирования алгоритмов, например в работе с условиями, счётчиками циклов, использованием локальных переменных и ресурсов;
- ошибки, связанные с неверной трактовкой данных, некорректной реализацией интерфейсов, совместимостью, производительностью и т. п., обычно пропускаются на уровне модульного тестирования и выявляются на более поздних стадиях.

Тестирование на основе потока управления — это стратегия структурного тестирования, которая использует поток управления программой в качестве модели.

Элементы, которые должны быть покрыты при прохождении тестов, определяются на основе структурных критериев тестирования C0, C1, C2. К

ним относятся вершины, дуги, пути управляющего графа программы (УГП), условия, комбинации условий и т. п..

К структурным критериям относятся:

- C0 (критерий тестирования команд) — набор тестов в совокупности должен обеспечить прохождение каждой команды. Это слабый критерий, который часто используется в больших программных системах, где другие критерии применить невозможно.
- C1 (критерий покрытия решений, ветвей) — тестирование всех возможных путей выполнения условных конструкций.
- C2 (критерий покрытия путей) — проверка выполнения всех путей через программу хотя бы один раз.
- Критерий покрытия условий — покрытие всех логических (булевских) условий в программе.

Критерии покрытия решений и условий не заменяют друг друга, поэтому на практике часто используется комбинированный критерий покрытия условий/решений, который совмещает требования по покрытию и решений, и условий.

Также применяются:

- критерий покрытия функций программы — каждая функция должна быть вызвана хотя бы один раз;
- критерий покрытия вызовов — каждый вызов каждой функции в программе должен быть осуществлён хотя бы один раз.

Тестирование на основе потока данных — метод, который направлен на выявление ссылок на неинициализированные переменные и избыточные присваивания (аномалий потока данных).

Элементы, которые должны быть покрыты, определяются на основе потока данных, то есть информационного графа программы.

Стратегия тестирования потока данных требует проверки всех взаимосвязей, включающих в себя ссылку (использование) и определение переменной, на которую указывает ссылка (то есть требуется покрытие дуг информационного графа программы).

Недостаток этого вида тестирования в том, что оно не включает критерий C1 и не гарантирует покрытия решений.

Существуют различные типы тестирования потока данных, например:

- All-Du-Paths — проверка всех возможных путей от определения переменной до каждого места её использования.
- All-Uses Testing — проверка каждого места, где используется переменная в приложении.
- All-Defs Testing — проверка каждого места, где переменная задаётся в коде.
- Testing for All-C-Uses — проверка всех возможных путей, в которых переменная используется в вычислениях.
- Testing for All-I-Uses — проверка всех путей, которые используют переменную, полученную из внешних входных данных.
- Testing for All-O-Uses — проверка всех путей, где переменная используется для генерации выходных данных.

Выбор подхода к тестированию зависит от целей, особенностей кода и требований к покрытию. Часто используется комбинация нескольких методов для повышения надёжности тестирования.