**Модульное и интеграционное тестирование. Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных. Организация модульного тестирования.**

**Особенности модульного тестирования**

Модульное тестирование (unit testing) направлено на проверку отдельных компонентов или модулей системы. Основные особенности:

Изолированное тестирование: Модули тестируются отдельно друг от друга, что помогает выявить ошибки на ранней стадии разработки.

Автоматизация: Часто используется автоматизация тестов, что позволяет быстро проводить регрессионное тестирование.

Использование заглушек: Для изоляции модулей применяются заглушки (stubs) и мок-объекты (mocks), заменяющие реальные зависимости.

Подходы к тестированию на основе потока управления и потока данных

**Тестирование на основе потока управления:**

Фокусируется на проверке различных путей выполнения программы, включающих условия, циклы и ветвления.

Используются методы, такие как покрытие условий и покрытие ветвей, чтобы гарантировать полное покрытие кода.

Тестирование на основе потока данных:

Внимание уделяется переменным и их состояниям в ходе выполнения программы.

Проверяются все пути данных, по которым переменные принимают значения и используются в выражениях.

**Организация модульного тестирования**

**Организация модульного тестирования включает несколько этапов:**

Подготовка тестовой среды: Создание необходимых тестовых данных, заглушек и мок-объектов.

Разработка тестовых случаев: Определение наборов тестов, покрывающих различные сценарии и условия.

Запуск тестов и анализ результатов: Проведение тестирования и анализ полученных результатов на предмет ошибок и проблем.

Регрессионное тестирование: Повторное тестирование после внесения изменений в код, чтобы убедиться, что исправления не вызвали новых ошибок.

Интеграционное тестирование

Интеграционное тестирование (integration testing) направлено на проверку взаимодействия между различными модулями или компонентами системы. Оно позволяет убедиться, что компоненты корректно работают вместе после их объединения.

**Особенности интеграционного тестирования:**

Иерархическое тестирование: Тестирование начинается с небольших групп модулей и постепенно расширяется до всей системы.

Использование драйверов и заглушек: Для изоляции отдельных компонентов используются специальные драйверы и заглушки, заменяющие недостающие компоненты.

Обнаружение интерфейсных ошибок: Интеграционное тестирование позволяет выявить ошибки в точках взаимодействия между модулями, которые могут не проявиться при модульном тестировании.

**Подходы к интеграционному тестированию:**

**Нисходящее тестирование:**

Тестирование начинается с самых верхних уровней системы и постепенно спускается к нижним.

Используются заглушки для имитации нижестоящих модулей, пока они не будут реализованы.

**Восходящее тестирование:**

Тестирование начинается с самых нижних уровней и постепенно поднимается к верхним.

Используются драйверы для имитации вышестоящих модулей, пока они не будут реализованы.

**Тестирование в виде "сандвича":**

Сочетает элементы нисходящего и восходящего тестирования.

Тестирование начинается одновременно с верхних и нижних уровней, постепенно сходясь к середине.

**Большой взрыв:**

Все модули интегрируются сразу, и затем проводится тестирование.

Это подход может быть рисковым, так как все ошибки приходится искать и исправлять одновременно.

**Организация интеграционного тестирования:**

Подготовка тестовой среды: Настройка окружения, включающего все необходимые компоненты и данные.

Разработка тестовых сценариев: Определение наборов тестов, охватывающих взаимодействие между модулями и системы в целом.

Запуск тестов и анализ результатов: Проведение тестирования и анализ полученных результатов на предмет ошибок и проблем взаимодействия.

Регрессионное тестирование: Повторное тестирование после внесения изменений, чтобы убедиться в отсутствии новых ошибок.

\_\_\_\_\_

Тестирование на основе потока управления - это метод тестирования, который основывается на проверке различных вариантов исполнения программы в зависимости от переходов в управляющих конструкциях (например, условных операторах или циклах).

Тестирование на основе потока данных - это метод тестирования, который фокусируется на проверке потока данных в программе

\_\_\_\_\_

**Разработать модульные тесты для программного модуля Validation, содержащий метод проверки валидности email. Код для программного модуля находится «Resorce/ModuleC#/». Оформить тест-кейсы для модульного тестирования.**