**Тестирование по методу «белого ящика».**

**Тестирование “белого ящика”** (или структурное тестирование, тестирование прозрачного ящика) - это метод тестирования программного обеспечения, при котором тестовые случаи разрабатываются на основе знания внутренней структуры кода, его алгоритмов и логики. Тестировщик имеет доступ к исходному коду и использует эти знания для создания тестов, которые проверяют различные пути выполнения кода, условия и циклы.

**Основные характеристики тестирования “белого ящика”:**

* Знание исходного кода: Тестировщик должен понимать структуру и логику кода.
* Проверка путей выполнения: Тесты разрабатываются для проверки различных путей выполнения кода, включая условные переходы, циклы и обработку исключений.
* Покрытие кода: Целью является достижение высокого покрытия кода тестами, то есть максимального процента строк кода, которые выполняются тестами.
* Методы: Используются различные методы, такие как покрытие операторов, покрытие решений, покрытие условий, покрытие путей.

**Цели тестирования “белого ящика”:**

* Проверить правильность логики и алгоритмов: Убедиться, что код выполняет правильные вычисления и принимает верные решения.
* Проверить обработку ошибок: Убедиться, что код правильно обрабатывает исключения и ошибки.
* Проверить все возможные пути выполнения кода: Убедиться, что все части кода выполняются хотя бы один раз.
* Оптимизировать код: Выявить узкие места в коде и предложить улучшения для повышения производительности.

**Преимущества тестирования “белого ящика”:**

* Позволяет выявить скрытые ошибки: Ошибки, которые трудно обнаружить с помощью тестирования “черного ящика”.
* Оптимизирует код: Помогает выявить неэффективные участки кода.
* Улучшает понимание кода: Тестировщик лучше понимает код, что облегчает выявление ошибок.

\_\_\_\_\_

**Разработать модульные тесты для программного модуля Serializer, содержащий метод сериализации объекта в Json. Код для программного модуля находится «Resorce/ModuleC#/». Оформить тест-кейсы для модульного тестирования.**