

# Создание тестов для тестовой системы в Arageli ЧЕРНОВОЙ ВАРИАНТ

С. С. Лялин

3 августа 2006 г.

При написании тестов нужно придерживаться некоторых единых естественных принципов. Это сделает тесты полезными не только для разработчика теста, но и для всех остальных текущих и последующих разработчиков библиотеки в будущем.

Здесь я просто приведу выжимку из переписки с тестовой группой ITLab, в течении которой мы пришли к некоторому единому стилю оформления тестов.

**Исключения.** Обрабатывайте все исключения, которые возникают при выполнении тестов. Вы, конечно, можете “отпустить” любое исключение, тогда его перехватит драйвер тестовой системы. Но, в данном случае полностью потеряется информация о том, а) что это было за исключение, б) при исполнении какой части вашего теста оно было сгенерировано и в) что за данные обрабатывались в момент исключения. А эта информация очень важна при дальнейшем выяснении источника ошибки.

Рекомендуется так писать тест, что бы все три вида информации, перечисленные выше, учитывались и выводились в выходной поток тестовой системы при возникновении исключения. Эту же информацию следует выводить при обнаружении тестом ошибки в Arageli, которая выражена не исключением.

Следует убедиться, что вы компилируете набор тестов при максимальном значении макроса `ARAGELI_DEBUG_LEVEL` (на данный момент это 3) и при определённом макросе `ARAGELI_ASSERT_THROW_EXCEPTION`. Первое условие обеспечит перехват всех непредвиденных ситуаций, а второе — то, что перехват будет осуществляться способом, полностью контролируемым приложением, т.е. при помощи исключений (контролируемый

ASSERT). Второй макрос определён в настройках проекта tests, входящем в arageli-all.sln.

Обратите внимание на то, что исключение `Arageli::exception` несёт в себе сообщение, которое так же нужно выводить в тестовый поток. В случае контролируемого ASSERT, исключение, которое генерируется в этом случае (`Arageli::assert_failed`, наследник `Arageli::exception`) содержит информацию о файле, номере строки и выражении, которое привело к исключению. Это очень полезная информация.

**Случайные данные.** В своих тестах вы можете использовать генераторы случайных объектов. При этом нужно помнить о том, что ваш тест должен быть полностью воспроизводим, т.е. не нужно инициализировать генератор случайных чисел, который лежит в основе ваших генераторов, текущим системным временем или каким-либо другим, неконтролируемым параметром. Но, это не означает, что генератор вообще не нужно инициализировать, что приведёт к одним и тем же результатам при различных вызовах теста и это никак нельзя будет изменить.

Сейчас тестовая система не имеет явного механизма передачи инициализационного значения, но вам следует так оформлять свои тесты, что бы они принимали один дополнительный параметр типа `int`, который вы можете использовать как начальное значение для псевдослучайных последовательностей (например, в функции `std::srand`). Когда интерфейс между тестом и драйвером тестовой системы будет реорганизован, то мы сможем передать этот параметр из драйвера вашему тесту. Сейчас же нужно инициализировать этот параметр самостоятельно нулевым значением. Да, это вызовет повторения при различных запусках вашего теста, но, это временное ограничение.

**Степень покрытия.** В своих тестах, которые основаны на том, что многократно повторяют одно и то же действие с разными псевдослучайными данными, нужно выбрать количество повторений. Конечно, можно зафиксировать число итераций, но это плохое решение, потому что никак не позволяет контролировать время работы вашего теста и степень покрытия исследуемого множества тестируемых объектов.

Сейчас ситуация в данном вопросе такая же как и в случае инициализационного значения. Вам нужно предусмотреть приём дополнительного параметра, который, в простейшем случае, может интерпретироваться как число итераций циклов внутри вашего теста. По возможности следует делать так, что бы время работы теста было линейным в зависимости от этого параметра. Условимся, что значение `ARAGELI_TESTSYS_COVERAGE_DEFAULT`, определённое в `tools/ts/ts.h`, будет значением по умолчанию для этого параметра (сейчас это 1000).

К тому же, если тест часто даёт сбой в одной и той же ситуации или

схожих ситуациях, то работу всего теста или его части следует прервать досрочно. Это делается из предположения о том, что информации, собранной на предыдущих итерациях теста, будет достаточно, что бы в дальнейшем обнаружить ошибку. Если этого не делать, то потом будет трудно работать с файлом вывода тестовой системы, так как он будет заполнен огромным количеством однотипной информацией, генерируемой на каждой итерации теста.