**Тест-туры - перевод оригинальных туров от Ольга Назиной (Киселевой)**

[Тур актера второго плана](http://okiseleva.blogspot.com/2015/11/supporting-actor-tour.html)

Этот тур поощряет тестировщиков фокусироваться на таких функциях, которые расположены на том же экране, что и функции, которые, как мы ожидаем, будет использовать большинство. Близость к главному функционалу увеличивает их «видимость». Будет ошибкой отдать им меньше внимания, чем они заслуживают.

[Тур глухого переулка](http://okiseleva.blogspot.com/2015/12/back-alley-tour.html)

Если ваша компания отслеживает использование функций, в этом туре вы сосредотачиваетесь на функциях в самом конце списка. Если вы отслеживаете покрытие кода, ваша задача — покрыть ранее непокрытый код.

[Тур полуночника](http://okiseleva.blogspot.com/2015/12/all-nighter-tour.html)

В исследовательном тестировании вопросы те же — может ли ваше приложение продержаться? Как долго оно может работать и обрабатывать данные перед тем, как развалится?

У приложения стресс и челендж. Потому что, если наращивать данные в памяти, постоянно читать с диска и писать на него— могут произойти плохие вещи. *Memory leaks, data corruption, race condition*… Вы узнаете много новых слов благодаря такому тесту!

[Тур коллекционера](http://okiseleva.blogspot.com/2016/02/collectors-tour.html)

Цель коллекционера — собрать полную коллекцию: бабочек каждого вида, каждого возраста. Аналогично и в тестировании — выбираем, что будем коллекционировать и собираем полный комплект.

[Тур одинокого бизнесмена](http://okiseleva.blogspot.com/2016/02/lonely-businessman-tour.html)

[Тур супермодели](http://okiseleva.blogspot.com/2016/03/supermodel-tour.html)

В этом туре вы должны думать поверхностно. Чтобы вы не делали, не погружайтесь глубоко. Этот тур не проверяет функции или процедуры, он проверяет то, как приложение выглядит и какое первое впечатление производит. Не фокусируйтесь на функциональности или реальном взаимодействии. Только на интерфейсе.

[Тур “второй бесплатно”](http://okiseleva.blogspot.com/2016/03/togof-tour.html)

TOGOF — простой тур, созданный только для тестирования многократно повторяющихся копий одного приложения, запущенных одновременно. Начните тур с запуска своего приложения, потом запустите вторую копию, потом третью. Теперь запустите на них функции, которые отжирают память или пишут на диск. Попробуйте использовать все разные копии для того, чтобы открыть один файл или одновременно передать данные в сеть. Возможно, они наткнутся друг на друга или сделают что-то неправильно, когда все попытаются прочитать один файл или записать туда что-то

[Тур шотландского паба](http://okiseleva.blogspot.com/2016/07/scottish-pub-tour.html)

Найти и протестировать функции, которые сложно найти, если заранее о них не знаешь. Эти функции передают “из уст в уста” наученные опытом коллеги, их можно нагуглить. Но с наскока простой пользователь их не найдет. Как житель мегаполиса пройдет мимо крутого паба, не зная, что внутри классное пиво и вежливые официанты.

[Тур, отмененный из-за дождя](http://okiseleva.blogspot.com/2015/04/rained-out-tour.html)

Идея тура в том, чтобы начать операцию и остановить ее.

Введите данные для поиска авиабилета, нажмите «Найти» и тут же отмените поиск. Отправьте документ на печать, подождите до середины процесса и отмените его.

Если программа делает что-то дольше пары секунд — отмените действие.

Если видите кнопку отмены — нажмите ее.

Способы отмены:

* *«Отмена»* или *«Назад»* в приложении
* *«Esc»* на клавиатуре
* *«Назад»* в браузере
* Shift + F4
* *«Х»* — крест в правом верхнем углу браузера

[Тур домоседа](http://okiseleva.blogspot.com/2016/08/couch-potato-tour.html)

Лениться! Делать так мало работы, как только возможно:

— соглашаться со всеми дефолтными значениями;

— оставлять поля пустыми;

— заполнять в форме минимум значений;

— никогда не кликать на “подробнее”;

— переключать вкладки без ввода данных;

— и так далее.

[Тур саботажника](http://okiseleva.blogspot.com/2017/03/saboteur-tour.html)

Цель тура

Саботировать работу приложения — запустить действие и отобрать ресурсы, необходимые для его выполнения.

[Тур по путеводителю](http://okiseleva.blogspot.com/2015/01/guidebook-tour.html)

Цель тура: найти все инструкции в приложении и выполнить их.

[Денежный тур](http://okiseleva.blogspot.com/2015/02/money-tour.html)

Тестировщик денежного тура:

* ходит на демонстрации,
* смотрит продающие продукт видеоролики,
* общается с Заказчиками.

Он знает, что показывают пользователям. И если новая функция разломала сценарий демонстрации, он укажет на проблему. Да, это может быть не самый интересный или критичный баг. Но исправьте его — и коллеги не будут краснеть на демонстрациях.

[Тур по ориентирам](http://okiseleva.blogspot.com/2015/05/landmark-tour.html)

*Метод Уиттакера, «дорога вперед»* — наметить список ориентиров и исследовать приложение, переходя от одного к другому.

[Интеллектуальный тур](http://okiseleva.blogspot.com/2015/06/intellectual-tour.html)

В рамках тестирования по туру мы задаем приложению сложные вопросы. Как мы можем заставить ПО работать так тяжело, как только возможно? Какие функции нагружают  приложение до предела? Какие входные значения и данные вызовут максимальную нагрузку? Какие входные значения могут ввести в заблуждение проверки на ошибочные данные? Какие внешние и внутренние данные приводят к специфичным результатам?

Конечно, задаваемые вопросы сильно зависят от нашего приложения:

[The FedEx Tour](http://okiseleva.blogspot.com/2015/06/the-fedex-tour.html)

Данные начинают свою жизнь как inputs (входные данные), сохраняются во внутренние переменные, изменяются и используются в вычислениях. Потом большинство данных доставляются как outputs (выходные данные) некоторому пользователю или части приложения. В FedEx-туре мы смотрим на эти данные.

[Внеурочный тур](http://okiseleva.blogspot.com/2015/06/after-hour-tour.html)

*Внеурочный тур* — проверить все задачи, выполняющиеся в фоновом режиме.

*Тур утренней дороги на работу* — проверить установку приложения.

[Тур сборщика мусора](http://okiseleva.blogspot.com/2015/07/garbage-collector-tour.html)

Выбрать цели (например, все пункты меню, все сообщения об ошибках, все диалоговые окна) и посещать каждый пункт наикратчайшим путем. Не останавливаясь тестировать детально, но замечая очевидные вещи.

[Тур по плохому району](http://okiseleva.blogspot.com/2015/07/bad-neighborhood-tour.html)

За месяц до начала разработки сложно сказать, какие районы кода будут «плохими». Чем сложнее функциональность, тем вероятнее ошибки. Чем слабее разработчик, тем вероятнее баги в его коде.

Используйте баг-трекер для анализа. Посмотрите — в каком компоненте больше всего ошибок? Там и ищите!

[Музейный тур](http://okiseleva.blogspot.com/2015/07/museum-tour.html)

Найти старый код в [системе контроля версий](https://ru.wikipedia.org/wiki/Система_управления_версиями), который:

— давно создан и больше не менялся;

— давно создан и недавно обновился.

И протестировать функциональность, заложенную в код.

[Тур предыдущей версии](http://okiseleva.blogspot.com/2015/09/prior-version-tour.html)

Регрессионное тестирование — проверяем, что функциональность, которая работала до новой версии, продолжает работать.