Лабораторные по КГ, группы Кострицкого А. С., 2025-02-08

Кажется, где-то эти правила или их аналоги читал Кирилл Леонидович.

1. Каждый модуль видит только свои зависимости и знает только о них.
2. Каждый модуль можно «продать» отдельно — он имеет смысл без модулей, которые уже зависят от него.
3. Подпрограмма (метод) знает только про себя и про свои зависимости, а не про контекст своего вызова: внутреннее наполнение должно быть видно из прототипа.
4. Любая подпрограмма безопасная и чистая.
5. Можно пользоваться любым фреймворком, позволяющим собрать stand-alone-приложение или набор скриптов.
6. Можно пользоваться только подпрограммами отображения точки (пиксе-ля), линии и многоугольника (полигона), кроме лабораторных, где сама цель в этом.
7. Старайтесь пользоваться сложными объектами и работать послойно, а не везде опускаться до координат точек.

Трансляция команд

Студенческий makefile для отображения команд преподавателя в команды студента: iu7-cg-labs-kostritsky.mk. Допускается любая передача управления дальше (в рамках разумного) — cmake, qmake, ms-build, etc.
Запуск из папки:

1. make -f iu7-cg-labs-kostritsky.mk run
   Запустить РЕЛИЗНОЕ приложение от лица пользователя.
2. … clean
   Очистить все промежуточные результаты сборки, КРОМЕ old-отчётов тестирования.
3. … release
   Собрать приложение для конечного пользователя. Имею право потом скопировать ready с содержимым на флешку, унести и запустить (при наличии зависимостей на целевой машине в пределах разумного — специально класть в папку рядом не надо). Это НЕ отладочная сборка, конечному пользователю в приложение отладочных флагов не насыпать.
4. …
   Провести всё модульное тестирование и поместить в файл. Если модульное тестирование не готово, очевидно, написать заглушку, честно создающую данный файл с отметкой о нулевом покрытии.
5. make -f iu7-cg-labs-kostritsky.mk func
   Аналогично, но для функционального тестирования. После функционального тестирования должна появляться папка ready/results с результатами тестирования — картинками или видео. Рядом с каждым результатом текстовый файл с описанием, что видим на картинке, генерируемый автоматически (потому что описание нужно создавать на этапе создания тестов).

Дополнительные команды и зависимости внутри makefile допускаются, а иногда вообще необходимы. Метод проверки качества написания makefile — по обратной ответственности.

Помните, что запуск может (но не всегда обязан) быть начат с команды clean.

Помните, что нельзя ставить в зависимость -old отчётам новые отчёты — старые отчёты всегда будут старше.

Модульное тестирование

Стопроцентное покрытие не ожидается, достаточно создать условия для тестирования и сделать один (два) теста. Метод проверки: преподавателем добавляется в одно конкретное место в программе один локализованный модульный тест, тесты проходят, итоговое покрытие увеличивается.

Функциональное тестирование

Тестируется не пользовательский интерфейс (use-cases), а математическая составляющая исходной задачи. . Ожидается не стопроцентное покрытие, а условия для лёгкого добавления нового теста и тестирования. Метод проверки тот же, что у модульного тестирования.

Для хранения исходных данных, включая описание теста, рекомендуется не разрозненная структура тестов (как в лабораторных по Си, где Вы вынуждены были это делать), а один json с последовательностью тестов. Из исходного json можно доставать данные с помощью утилиты jq. И поля будут как в словаре.