

1. Unit testing. File Storage

Задание	Task 1. Unit Testing (file storage)
Цель задания	Изучение инструментов модульного тестирования и применение полученных знаний на практике
Язык	Java, Python, C#
Тип задания	Автоматизация тестирования
Необходимые компетенции и техническая подготовка	Коммуникация; Знание ООП; Использование фреймворков/библиотек для тестирования TestNG/PyTest/NUnit
Описание задания	Необходимо разработать модульные тесты для приложения File Storage: Java File Storage.rar C# File Storage.rar TBU: Python FileStorage.rar
Время выполнения	12 ч
Требования к выполнению задания	<ol style="list-style-type: none">Исходный код тестируемого приложения при помощи механизмов библиотеки TestNG/PyTest/NUnitТесты, реализующие одну логику проверки (вызывают последовательность одних и тех же действий), но выполняющиеся на разных наборах данных, должны быть параметризованы.Проект должен собираться автоматически. Зависимости/используемый библиотеки должны быть указаны в pom (Java)/requirements.txt(Python)/NuGet dependencies (C#) файлах.Проект должен запускаться с помощью maven(Java)/pytest(Python) команды.Тесты должны строиться с учетом документации к методам в коде тестируемого приложения.Каждый тест должен содержать одну конкретную проверку, чтобы было четко и ясно понятно по какой причине упал тест.Тестовое покрытие должно быть максимально полным:<ul style="list-style-type: none">тестирование должно производиться на всех возможных комбинациях данных (однако данные должны различаться и потенциально выявлять различные дефекты либо сужать/конкретизировать по какой причине, в каких случаях метод работает некорректно, а в каких возможно корректно)один и тот же дефект может и должен в большинстве случаев обнаруживаться тестами разных методов (если они ссылаются друг на друга или обращаются к одним данным)Код тестов должен соответствовать принятым в компании Coding GuidelinesКод задания необходимо поместить в отдельную ветку (Branch) репозитория студента. Branch name = t1_unit-tests-fs.Найденные тестами дефекты в приложении необходимо описать в файле BugList. Файл положить в корень ветки с кодом задания.
Ожидаемый результат выполнения задания	<ol style="list-style-type: none">Ветка в Git, содержащая код проекта с тестами, а также файл со списком найденных дефектовПисьмо куратору и техническому экспертам, содержащее:<ul style="list-style-type: none">Ссылку на Pull/Merge RequestДля Java/Python: консольная команда для запуска тестов (через maven / pytest)Список найденных дефектовВремя, затраченное на выполнение заданияНа что было затрачено время сверх оценки времени на задание, если было превышениеЧто вызвало наибольшие сложности при выполнении заданияКакие остались нерешенные проблемы, если осталисьТема письма = TCXXX. <student (e.g. v.pupkin)>. Task 1: Unit Testing Results