УНИВЕРСИТЕТ ИТМО

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
“Национальный исследовательский университет ИТМО”

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине
“Методы и средства программной инженерии”
Тема: «Apache ant».

Вариант: 334541.

выполнили:
Студенты групп P32131, P32092
Коновалов Арсений Антонович
Голиков Андрей Сергеевич
Преподаватель:
Бострикова Дарья Константиновна

г. Санкт-Петербург
2023 г.

1 Задание

Сконфигурировать в своём домашнем каталоге репозитории svn и git и загрузить в них начальную ревизию файлов с исходными кодами (в соответствии с выданным вариантом).

Воспроизвести последовательность команд для систем контроля версий svn и git, осуществляющих операции над исходным кодом, приведённые на блок-схеме.

При составлении последовательности команд необходимо учитывать следующие условия:

Цвет элементов схемы указывает на пользователя, совершившего действие (красный - первый, синий - второй). Цифры над узлами - номер ревизии. Ревизии создаются последовательно. Необходимо разрешать конфликты между версиями, если они возникают.

Лабораторная работа #3

Вариант 334541

Внимание! У разных вариантов разный текст задания!

Написать сценарий для утилиты Apache Ant, реализующий компиляцию, тестирование и упаковку в jar-архив кода проекта из лабораторной работы №3 по дисциплине "Веб-программирование".

Каждый этап должен быть выделен в отдельный блок сценария; все переменные и константы, используемые в сценарии, должны быть вынесены в отдельный файл параметров; MANIFEST.MF должен содержать информацию о версии и о запускаемом классе.

Сценарий должен реализовывать следующие цели (targets):

1. **compile** -- компиляция исходных кодов проекта.
2. **build** -- компиляция исходных кодов проекта и их упаковка в исполняемый jar-архив. Компиляцию исходных кодов реализовать посредством вызова цели **compile**.
3. **clean** -- удаление скомпилированных классов проекта и всех временных файлов (если они есть).
4. **test** -- запуск junit-тестов проекта. Перед запуском тестов необходимо осуществить сборку проекта (цель **build**).
5. **doc** - добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта, а также генерация и добавление в архив javadoc по всем классам проекта.
6. **native2ascii** - преобразование native2ascii для копий файлов локализации (для тестирования сценария все строковые параметры необходимо вынести из классов в файлы локализации).
7. **music** - воспроизведение музыки по завершению сборки (цель **build**).
8. **diff** - осуществляет проверку состояния рабочей копии, и, если изменения не касаются классов, указанных в файле параметров выполняет commit в репозиторий git.
9. **report** - в случае успешного прохождения тестов сохраняет отчет junit в формате xml, добавляет его в репозиторий git и выполняет commit.
10. **env** - осуществляет сборку и запуск программы в альтернативных окружениях; окружение задается версией java и набором аргументов виртуальной машины в файле параметров.

Вопросы к защите лабораторной работы:

1. Тестирование ПО. Цель тестирования, виды тестирования.
2. Модульное тестирование, основные принципы и используемые подходы.
3. Пакет JUnit, основные API.
4. Системы автоматической сборки. Назначение, принципы работы, примеры систем.
5. Утилита make. Make-файлы, цели и правила.
6. Утилита Ant. Сценарии сборки, цели и команды.

2 Выполнение

2.1 Git

`https://github.com/FooolyHARD/MSP-3`

3 Вывод

Выполняя лабараторную работу мы 9 часов подряд долбились со всяким древним говном и ставили УННВ на звук завершения компиляции проекта.