

МИНИСТЕРСТВО НАУКИ И ОБРАЗОВАНИЯ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине
ОСНОВЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

Вариант №-432321

Выполнили:

Студенты группы Р3232

Чмурова Мария Владиславовна

Комягин Дмитрий Анатольевич

Проверила:

Бострикова Дарья

Константиновна

г. Санкт-Петербург

2024 год

Оглавление

Задание	3
Код Apache Ant (build.xml)	5
Код build.properties	6
Вывод	7

Задание

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 запуске класса.

Сценарий должен реализовывать следующие цели (targets):

`compile` -- компиляция исходных кодов проекта.

`build` -- компиляция исходных кодов проекта и их упаковка в исполняемый jar-архив. Компиляцию исходных кодов реализовать посредством вызова цели `compile`.

`clean` -- удаление скомпилированных классов проекта и всех временных файлов (если они есть).

`test` -- запуск junit-тестов проекта. Перед запуском тестов необходимо осуществить сборку проекта (цель `build`).

`music` - воспроизведение музыки по завершению сборки (цель `build`).

`doc` - добавление в MANIFEST.MF MD5 и SHA-1 файлов проекта, а также генерация и добавление в архив javadoc по всем классам проекта.

`native2ascii` - преобразование `native2ascii` для копий файлов локализации (для тестирования сценария все строковые параметры необходимо вынести из классов в файлы локализации).

`xml` - валидация всех xml-файлов в проекте.

`env` - осуществляет сборку и запуск программы в альтернативных окружениях; окружение задается версией java и набором аргументов виртуальной машины в файле параметров.

`team` - осуществляет получение из git-репозитория 4 предыдущих ревизий, их сборку (по аналогии с основной) и упаковку получившихся jar-файлов в zip-архив. Сборку реализовать посредством вызова цели `build`.

alt - создаёт альтернативную версию программы с измененными именами переменных и классов (используя задание replace/replaceregexp в файлах параметров) и упаковывает её в jar-архив. Для создания jar-архива использует цель build.

history - если проект не удаётся скомпилировать (цель compile), загружается предыдущая версия из репозитория git. Операция повторяется до тех пор, пока проект не удастся собрать, либо не будет получена самая первая ревизия из репозитория. Если такая ревизия найдена, то формируется файл, содержащий результат операции diff для всех файлов, изменённых в ревизии, следующей непосредственно за последней работающей.

Код Apache Ant (build.xml)

Ссылка на GitHub с файлом build.xml:

https://github.com/kkettch/lab_3/blob/master/build.xml

Код build.properties

Ссылка на GitHub с файлом build.properties:

https://github.com/kkettch/lab_3/blob/master/build.properties

Вывод

В ходе лабораторной работы мы составили собственный сценарий сборки проекта java при помощи утилиты Apache Ant. Также написали собственные цели различных направленностей. Узнали и о других утилитах сборки, их отличиях между собой.