Домашнее задание по теме «Работа с Maven. JAR»

Формулировка задания:

1. Преобразовать проект для Промежуточной аттестации 1 в maven-проект.

- Создаем pom.xml файл в проекте с домашним заданием.
- Пишем/изменяем теги заголовка groupId, artifactId.
- Добавить зависимости
- Добавляем тег <build>. Указываем в нем где находятся ресурсы, имя результирующего jar файла.

2. Настроить зависимости для проекта в dependencies Maven.

3. Собрать исполняемый архив средствами Maven со сторонними библиотеками (JUnit и других библиотек, которые включены в проект Промежуточной аттестации 1).

4. Обратить внимание на MANIFEST-файл и его настройку с классами сторонней библиотеки.

5. Отдельно показать в отчёте (сделать скрины) собранный JAR файл через Maven.

Программа реализуется в отдельной ветке git homeworks/homework16 с описанием хода работы по задаче.

В корне папки с программой должен быть файл .gitignore.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

Дополнительное задание:

1. Создайте проект на базе Maven.

2. Добавьте в проект JUnit Jupiter & Mockito

3. Создайте сервисный класс BonusService со следующим исходным кодом:

```
public class BonusService {
    public long calculate(long amount, boolean registered) {
            intpercent = registered ? 3 : 1;
            long bonus = amount * percent / 100;
            longlimit = 500;
            if (bonus > limit) {
                    bonus = limit;
            }
            return bonus;
    }
}
```

4. Протестируйте сервисный класс с помощью unit-тестов Junit и Mockito.

5. Запустите тесты через mvn clean test, убедитесь что они запускаются и проходят.

6. Проверьте покрытие тестами.

7. Убедитесь, что тесты запускаются и проходят.

8. Соберите JAR архив приложения через Maven с учетом сторонних библиотек.

Программа реализуется в отдельной ветке git homeworks/homework16Addition с описанием хода работы по задаче.

В корне папки с программой должен быть файл .gitignore.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

Планируемый результат:

Реализован архив, содержащий MANIFEST-файл с классами сторонней библиотеки. В репозитории .m2 наличие собственного архива-библиотеки.

Описания плана работы:

1. Ссылка на программу в репозитории github;
2. Отчёт со скринами выполнения задач - постановка задачи, код задачи и результат в консоли Intellij Idea. Подготовка необходимых class-файлов и последующая упаковка в архивы с помощью фаз Maven.

Перечень инструментов, необходимых для реализации деятельности:

Персональный компьютер, JDK 17 (либо OpenJDK 17), Intellij Idea для разработки на Java, GIT, Tortoise GIT, jUnit 5.8.1, Apache Maven 3.9.5