1. Вам необходимо выстроить и написать архитектуру для банковских продуктов. Банковскими продуктами являются – кредитная карта, дебетовая карта, валютная дебетовая карта и вклад.

- Карты и вклад имеют параметры: валюта, баланс и название

- Карты содержат методы: пополнение, списание, запрос баланса

- Кредитная карта имеет дополнительный метод запрос задолженности и дополнительный параметр процентная ставка

- Вклад содержит методы: пополнение, запрос баланса, закрытие

Опишите вашу структуру/архитектуру в классах. Учтите, что продуктов в будущем станет больше - то есть предусмотрите возможность изменения вашей архитектуры без различных проблем и лишних правок.

Напишите unit тесты для проверки реализованной архитектуры.

2.

Зарегистрироваться на сервисе передачи данных о погоде - <https://weatherstack.com/>. Бесплатный аккаунт 1000 вызовов в месяц. Поэтому будьте внимательней - должны остаться варианты для запуска во время проверки выполнения задания. Документация по API – <https://weatherstack.com/documentation>.

Используя любой BDD фреймворк реализовать:

**Позитивный тест:**

1. Запросить текущую погоду минимум по четырем городам на свой выбор.
2. Распарсить результат, сравнить с ожидаемыми значениями из тестового набора (language, location и т.п.). Ожидаемые тестовые данные можно определить или задать для каждого города корректные (localtime, utc\_offset и т.п.), либо можно задать\сгенерировать случайным образом с соблюдением формата (wind\_speed, temperature и пр.).
3. Вывести расхождения по результату сравнения по каждому значению в лог.

**Негативный тест:**

Получить 4 варианта ошибок из списка API Errors (на выбор), сравнить с ожидаемым результатом.

Результат выполнения тестов должен быть в отчете Allure.