**Курсовой проект по курсу «Основы backend-разработки»**

**Код для виджета «Операции по счетам»**

👨🏻‍💻 IT-отдел крупного банка делает новую фичу для личного кабинета клиента. Это виджет, который показывает несколько последних успешных банковских операций клиента. Вам доверили реализовать алгоритм, который на бэкенде будет готовить данные для отображения в новом виджете.

**Задача**

Реализуйте функцию, которая выводит на экран список из 5 последних выполненных клиентом операций в формате:

<дата перевода> <описание перевода>

<откуда> -> <куда>

<сумма перевода> <валюта>

# Пример вывода для одной операции:

14.10.2018 Перевод организации

Visa Platinum 7000 79\*\* \*\*\*\* 6361 -> Счет \*\*9638

82771.72 руб.

**Требования**

* Последние 5 выполненных (EXECUTED) операций выведены на экран.
* Операции разделены пустой строкой.
* Дата перевода представлена в формате ДД.ММ.ГГГГ (пример: 14.10.2018).
* Сверху списка находятся самые последние операции (по дате).
* Номер карты замаскирован и не отображается целиком в формате XXXX XX\*\* \*\*\*\* XXXX (видны первые 6 цифр и последние 4, разбито по блокам по 4 цифры, разделенных пробелом).
* Номер счета замаскирован и не отображается целиком в формате \*\*XXXX (видны только последние 4 цифры номера счета).

**Данные**

Файл со списком операций, совершенных клиентом банка:

[operations.zip](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/dd686a1e-f5aa-4c73-b4f2-163a93b8432b/operations.zip)

💡 Скачайте файл и положите его в проект. Работать с ним надо через библиотеку json. Файл модифицировать нельзя. Все нестандартные записи обрабатывать программно.

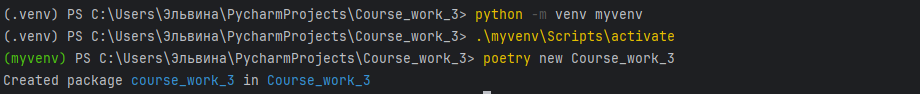
По каждой операции есть следующие данные:

* id — id транзакциии
* date — информация о дате совершения операции
* state — статус перевода:
  + EXECUTED — выполнена,
  + CANCELED — отменена.
* operationAmount — сумма операции и валюта
* description — описание типа перевода
* from — откуда (может отсутстовать)
* to — куда

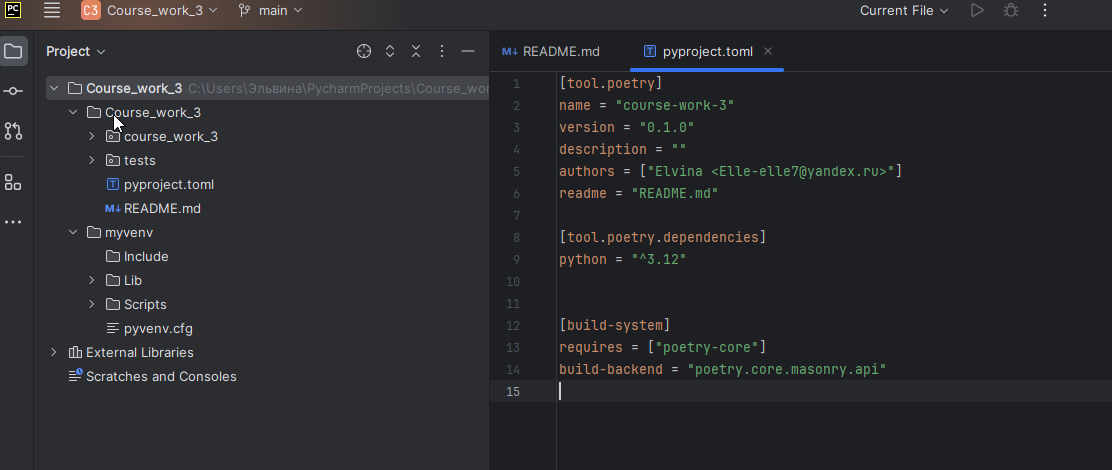
**Критерии выполнения**

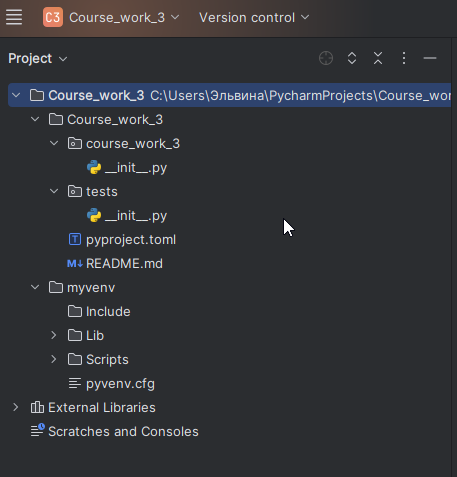
* [ ] Проект выложили на GitHub.
* [ ] Есть файл .gitignore
* [ ] Оформили файл [README.md](http://README.md).
* [ ] В проекте есть минимум 2 ветки, причем разработка ведется в develop, а стабильная версия на момент сдачи проекта лежит в ветке main.
* [ ] Разработка проекта ведется в виртуальном окружении, в проекте есть информация о требованиях проекта (зависимостях).
* [ ] К проекту написали тесты с покрытием не менее 80%.
* [ ] Тесты написали на pytest или unittest.
* [ ] Проект структурированный, читаемый, каждая функция — не более 50 строк.
* [ ] Работа с файлом ведется через библиотеку json

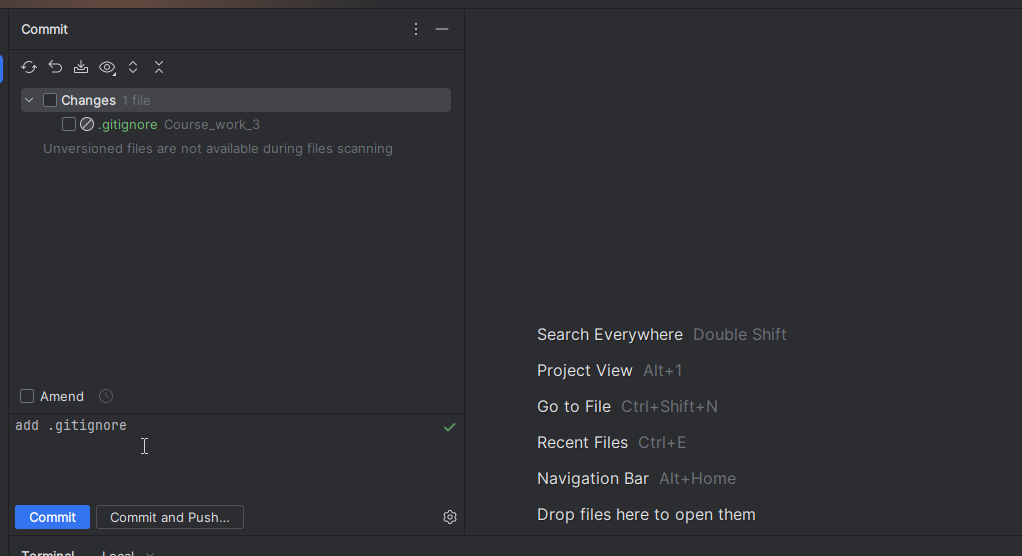
**Решение**





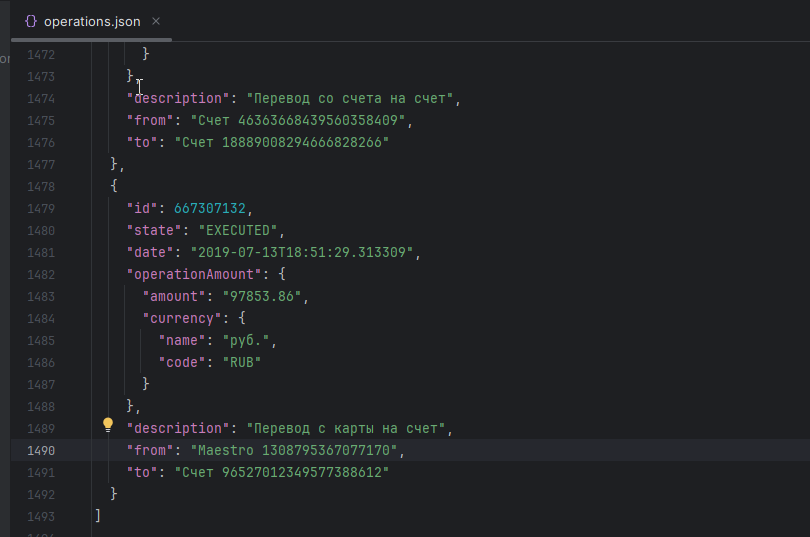


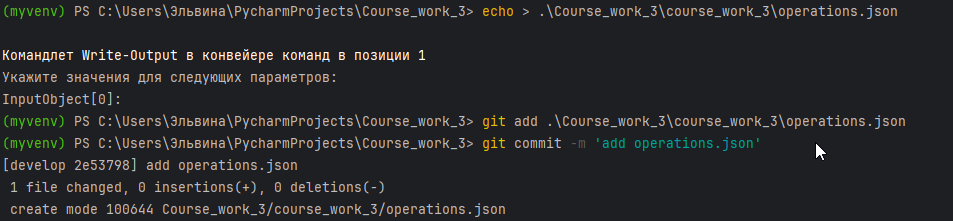


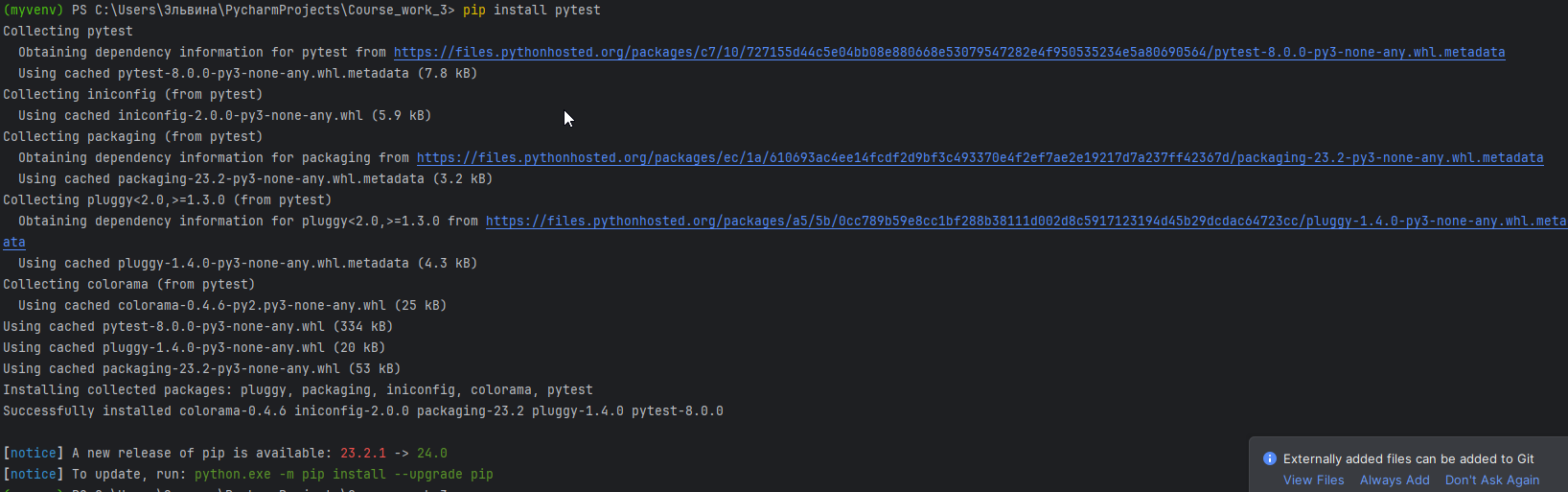


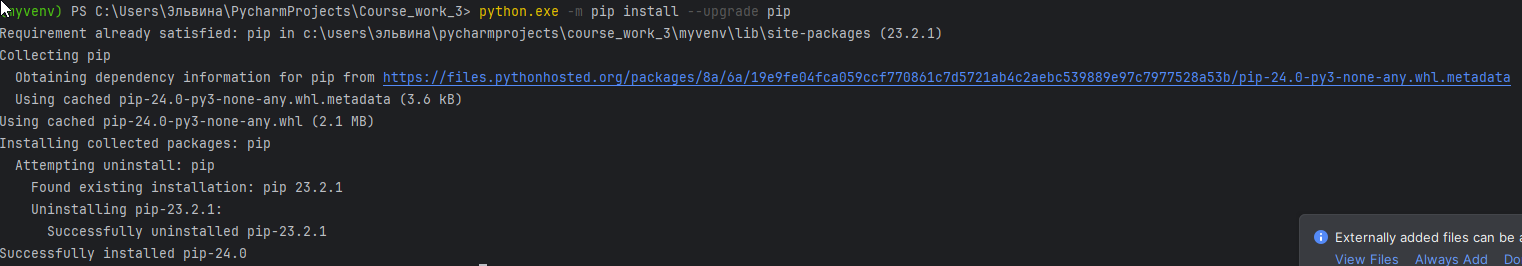
**После чего создаем ветку девелоп и переключаемся на нее.**

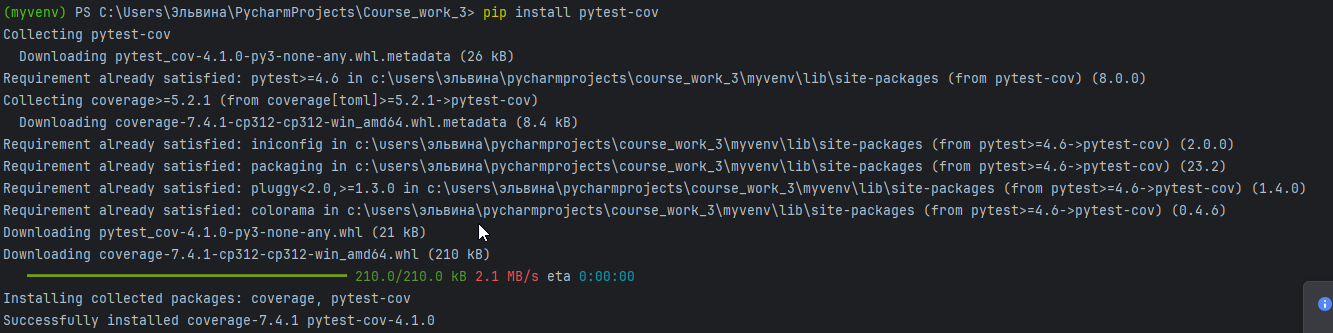
**Потом создаем файл джейсон в этой ветке, добавляем и коммитим**

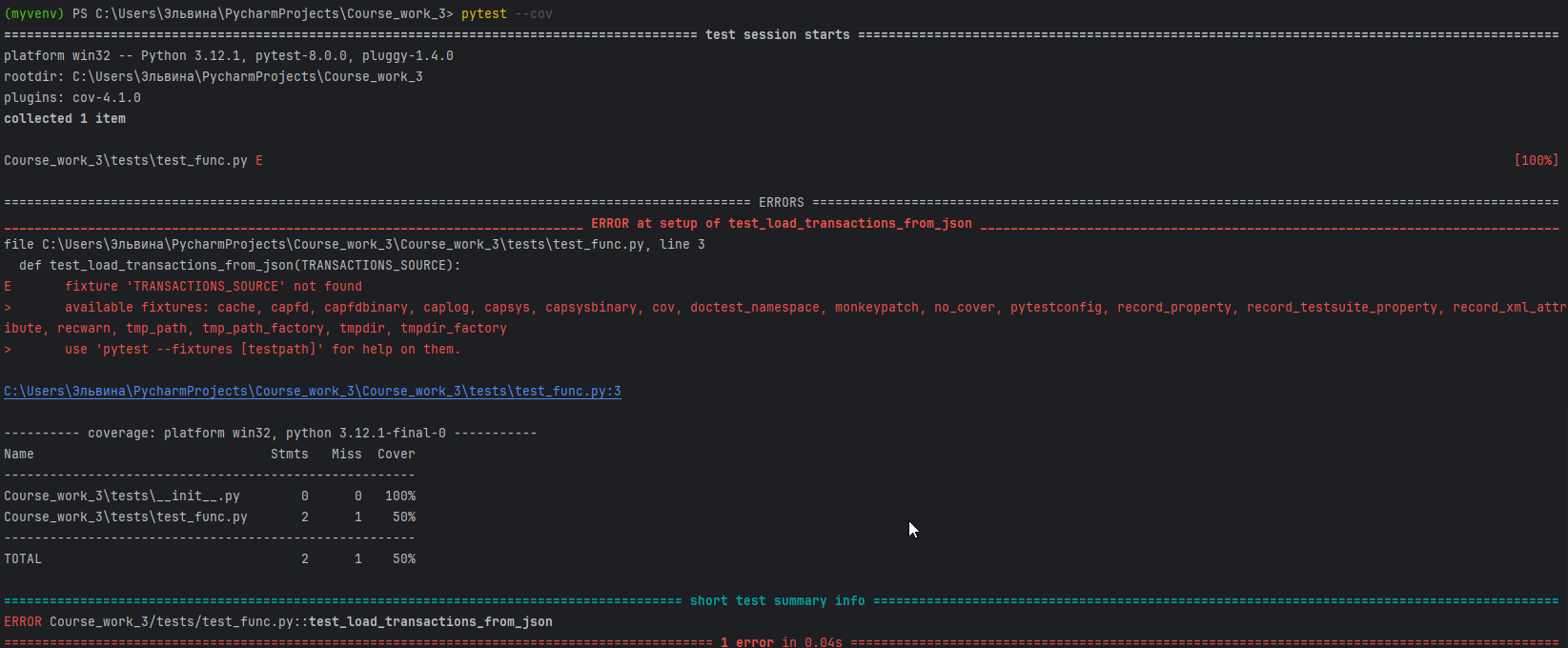




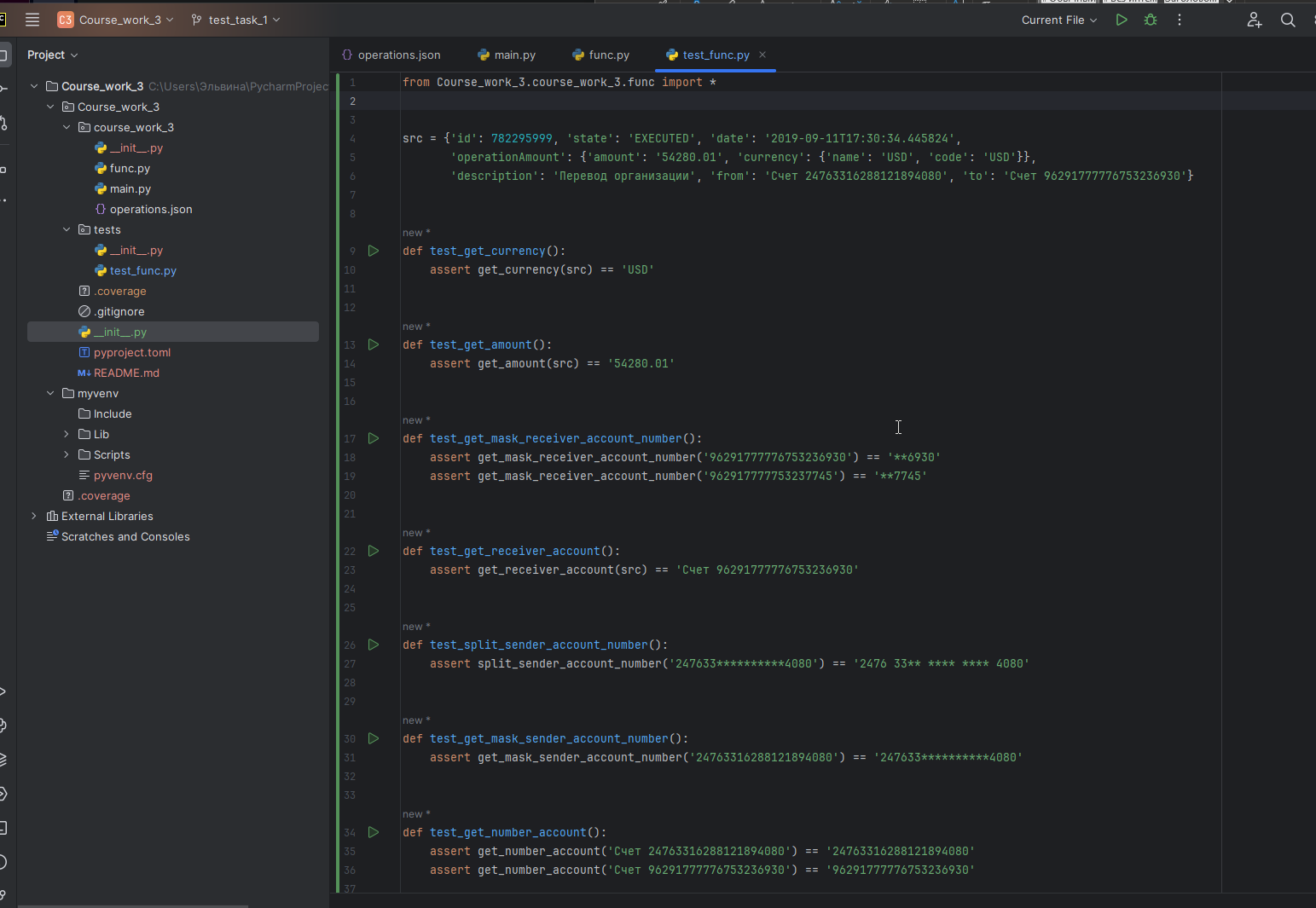


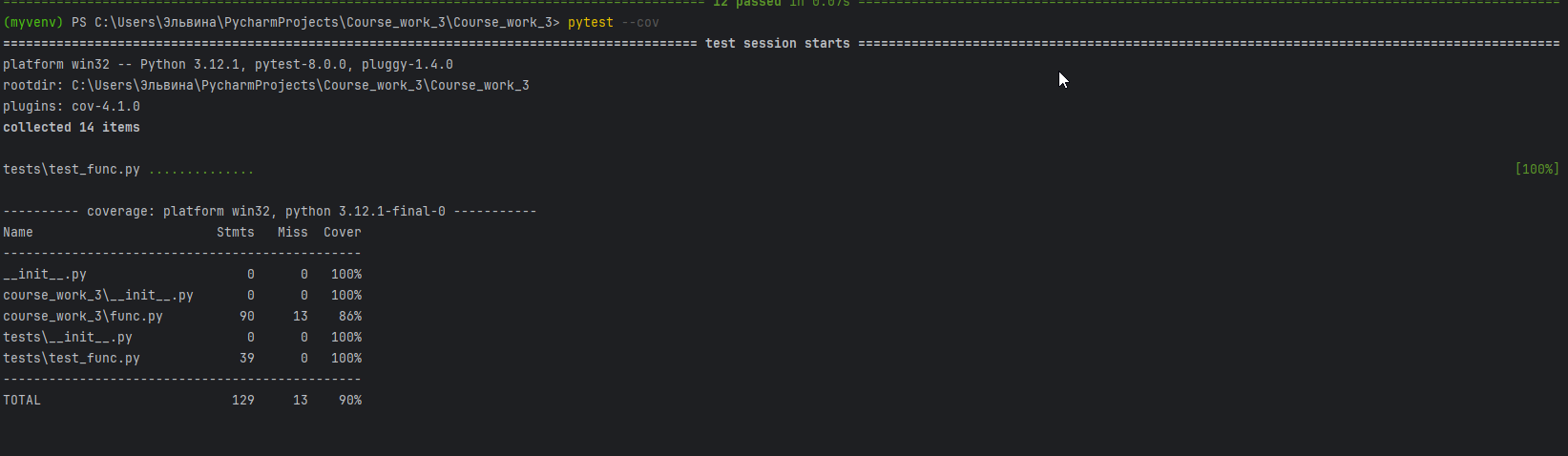


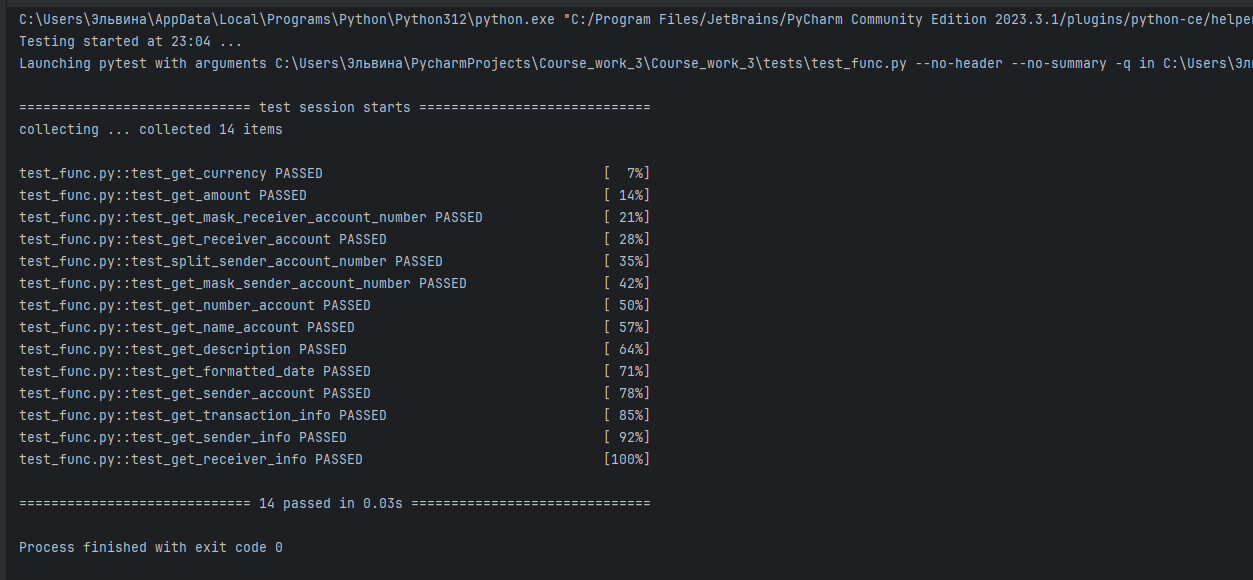




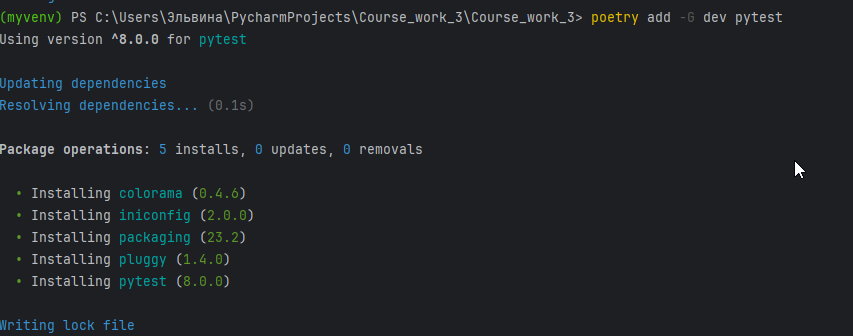
**Запуск тестов**

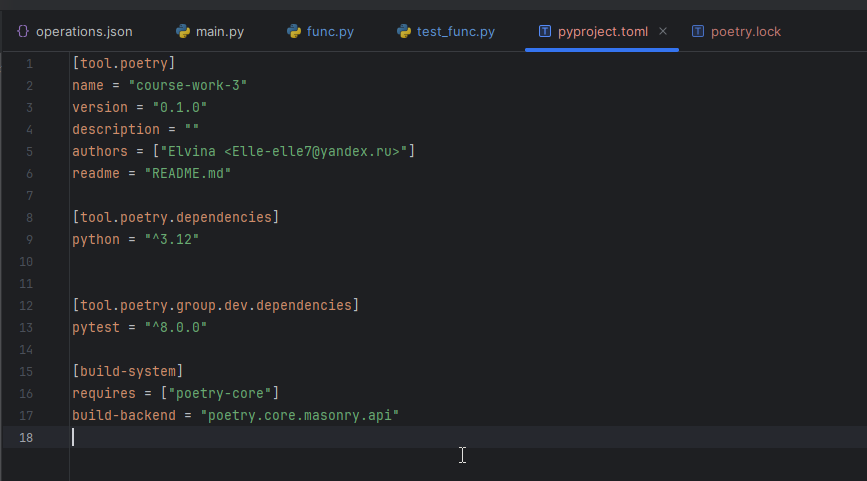






**Установим зависимости**





[**https://github.com/ElvinaVen/Course\_work\_3**](https://github.com/ElvinaVen/Course_work_3)

