**Условия домашки**

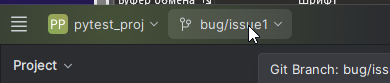
Вам передали проект, над которым вел работу другой программист. Теперь вам предстоит разобраться в существующем коде и дописать к нему тесты. Дополнительно реализуйте новый функционал и покройте его тестами.

**Критерии выполнения домашнего задания**

* На GitHub есть два репозитория, в одном — решения с pytest, в другом — c unittest.
* Каждое задание сделали в своей ветке, которые смерджены в ветку develop.

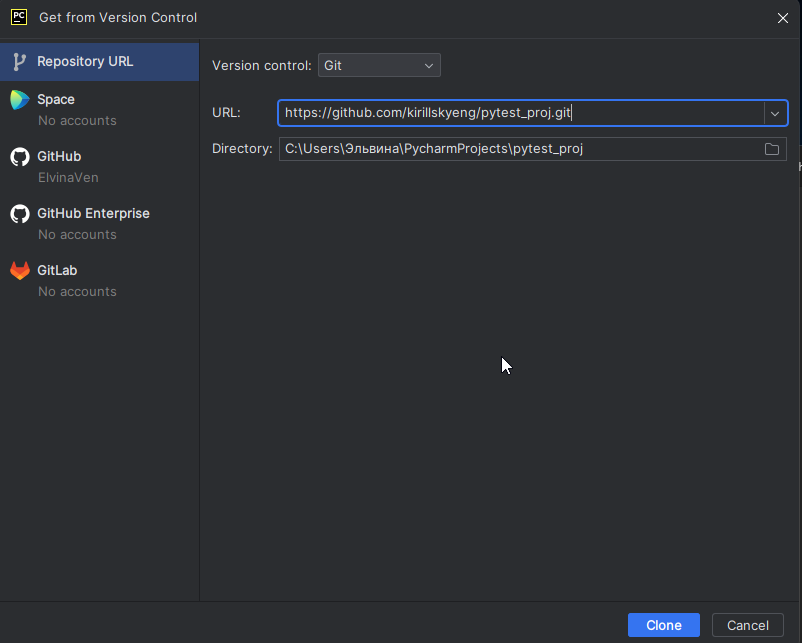
**Задание 1. Фикс тестов**

Работу по этой задаче ведите через PyCharm в ветке bug/issue1.



**Выполните следующие шаги:**

1. Клонируйте проект: <https://github.com/kirillskyeng/pytest_proj>



Убедитесь, что вы получили из удаленного репозитория все ветки и переключились на ветку разработки develop, иначе директория tests не будет видна.

Вам помогут пункты Pull и New Branches меню Git в PyCharm или git-команды:

* + git pull --all
  + git branch -a
  + git checkout develop

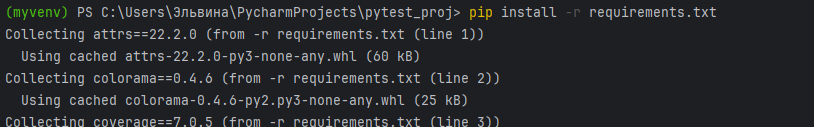
Выполните упражнение в новой ветке

1. Настройте виртуальное окружение.





1. Установите зависимости.



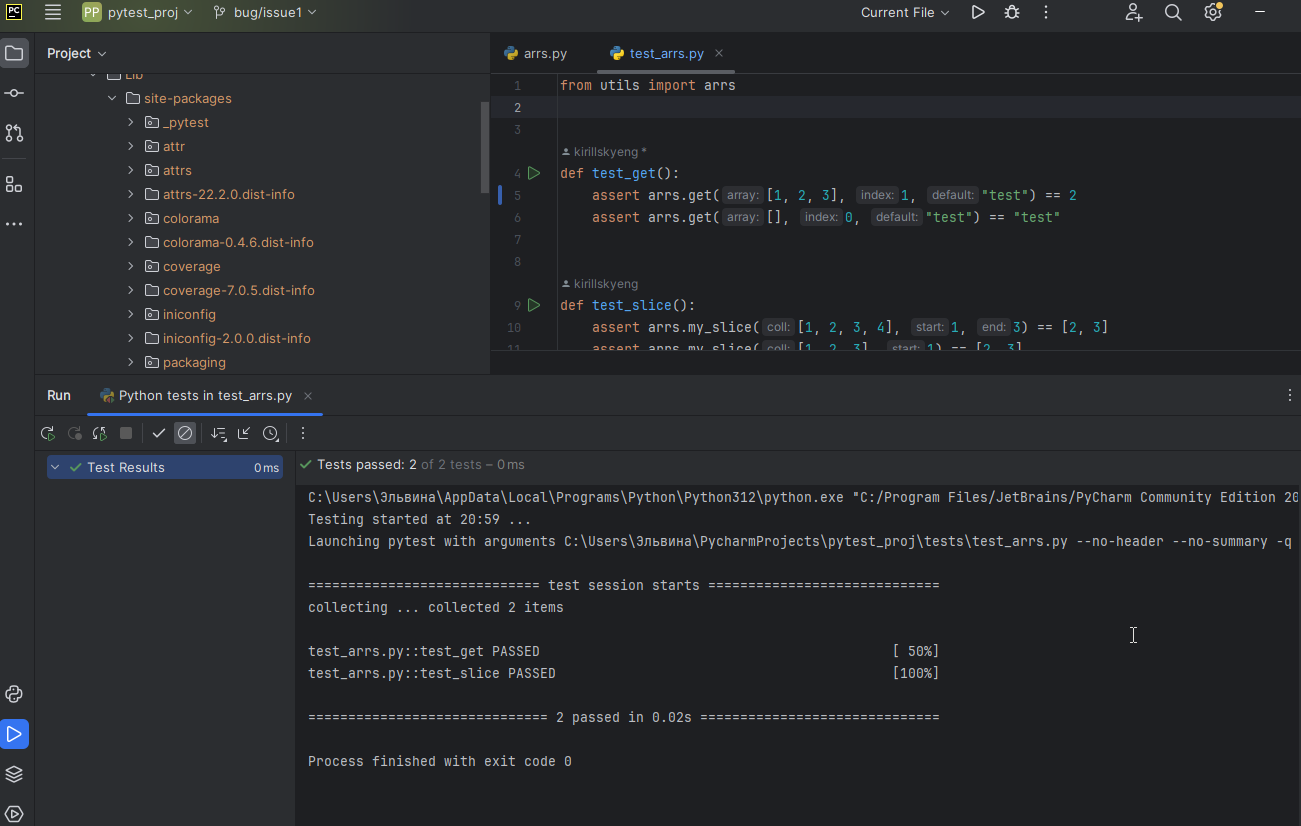
1. Разберитесь со структурой проекта. В проекте есть следующие директории:
   * utils

 — содержит модули для работы с массивами;

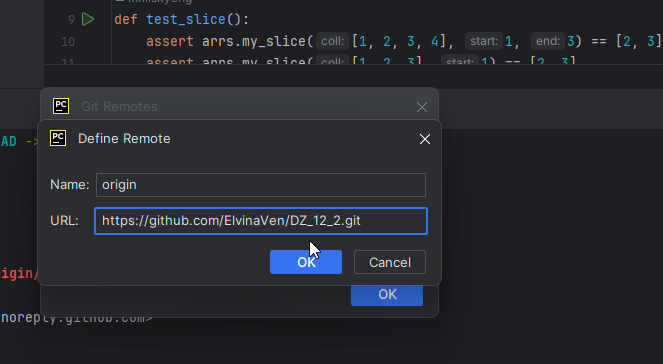
* + tests

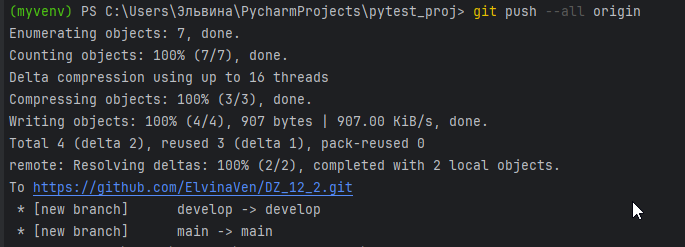
 — содержит файлы с тестами.

1. Добейтесь того, чтобы тесты, написанные к модулю arrs, проходили. Тесты пишите с помощью pytest.

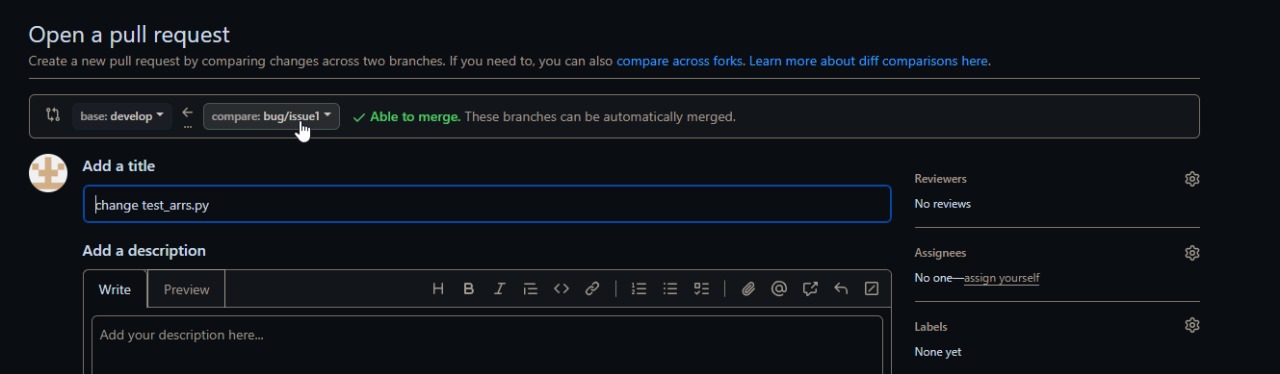


1. Залейте проект **себе** на GitHub, создав новый репозиторий. Чтобы отправить в удаленный репозиторий все ветки сразу, используйте команду git push --all origin.

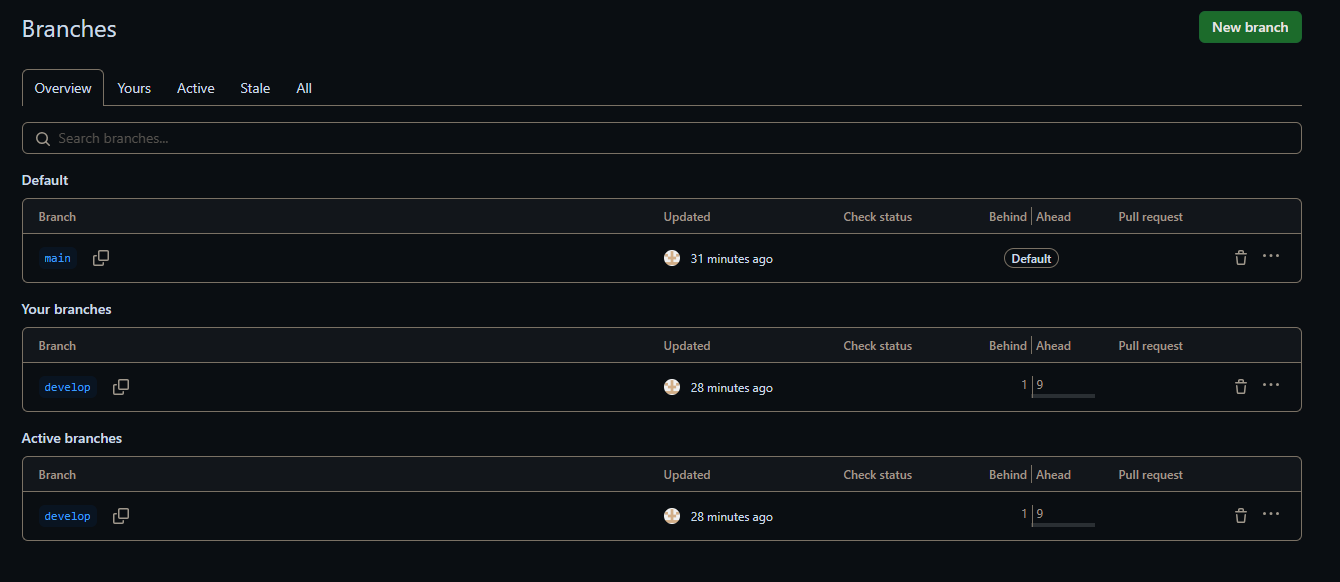




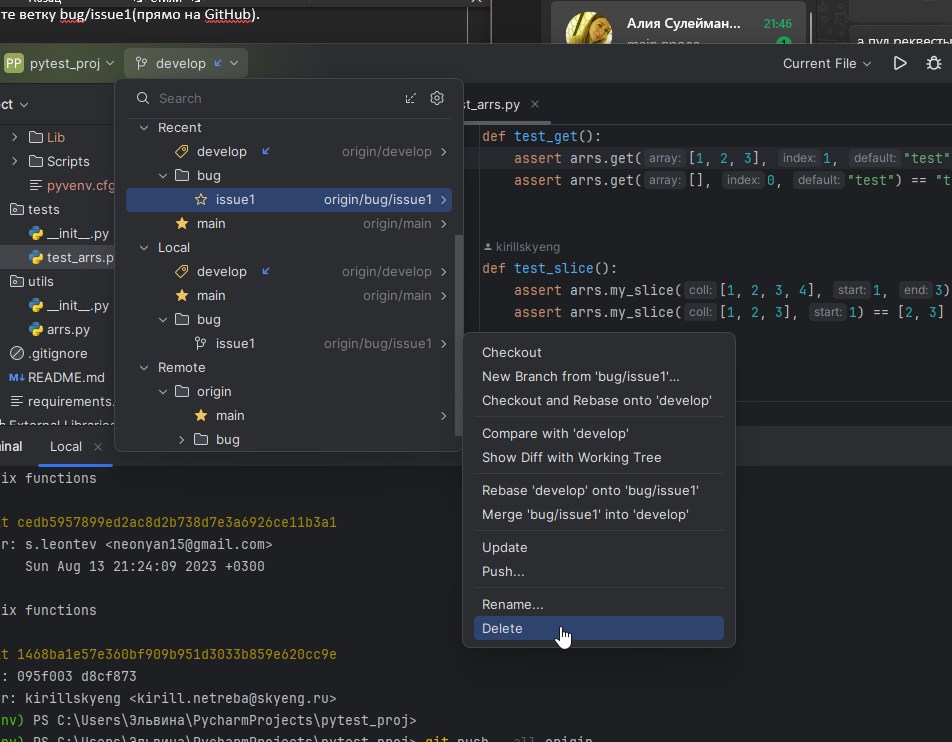
1. Сделайте pull request в ветку develop проделанной в задании работы и за тимлида заапрувьте его.

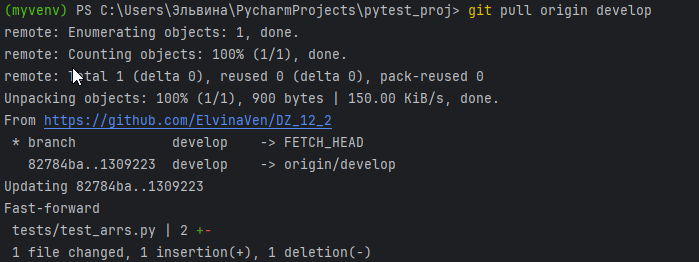


1. После апрува пул-реквеста удалите ветку bug/issue1(прямо на GitHub).



1. Удалите локальную ветку bug/issue1 и обновите ветку develop.



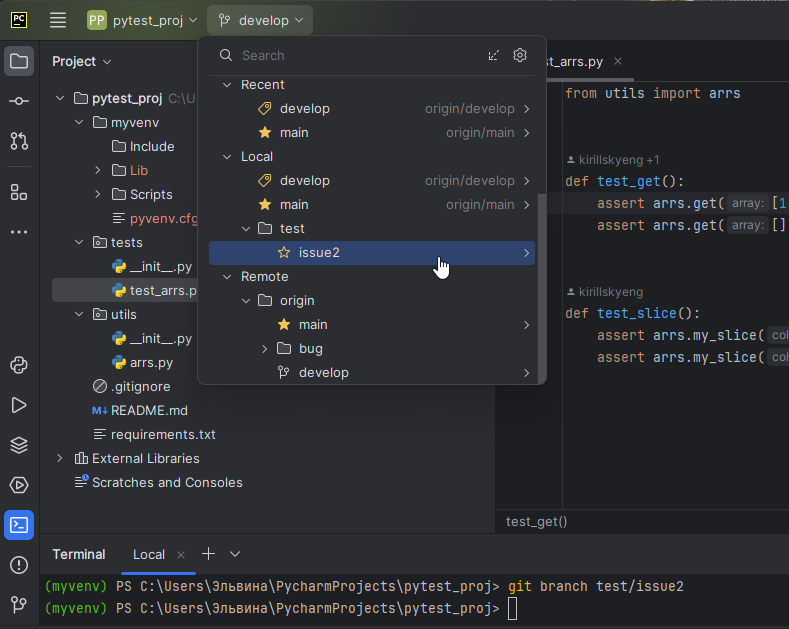


Итог



**Задание 2. Покрытие тестами**

Работу по этой задаче ведите в ветке test/issue2.



**Выполните следующие шаги:**

1. Узнайте процент покрытия тестами в проекте.

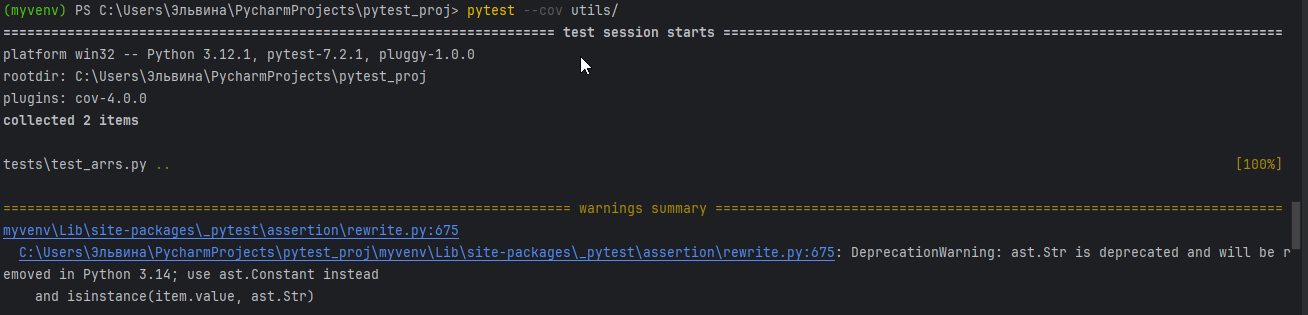
Кнопка для запуска оценки покрытия тестами работает в PyCharm только в профессиональной версии.

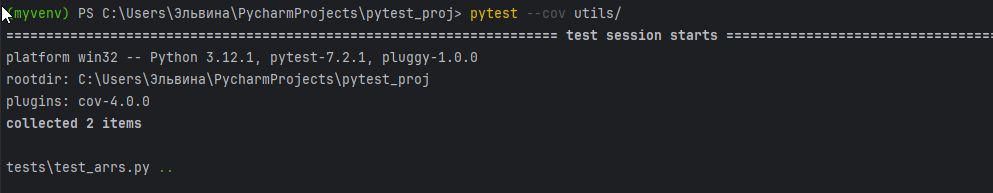
Чтобы запустить покрытие тестами, выполните в терминале, находясь в виртуальном окружении проекта, команду:

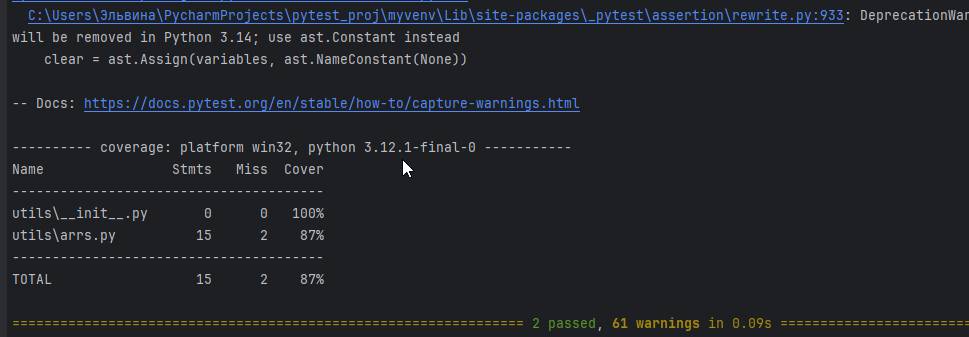
pytest --cov utils/

Или для генерации HTML-отчета:

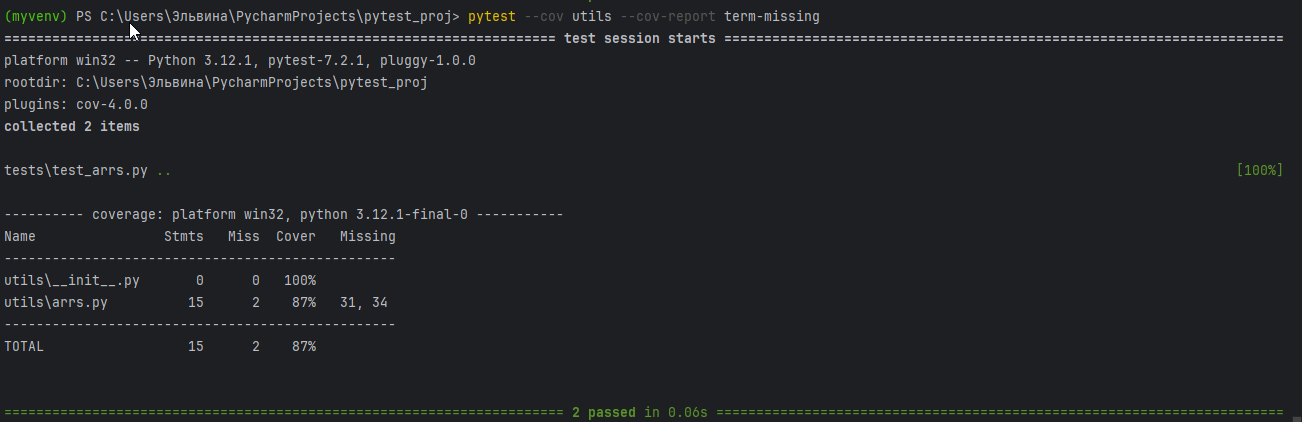
pytest --cov=utils --cov-report=html



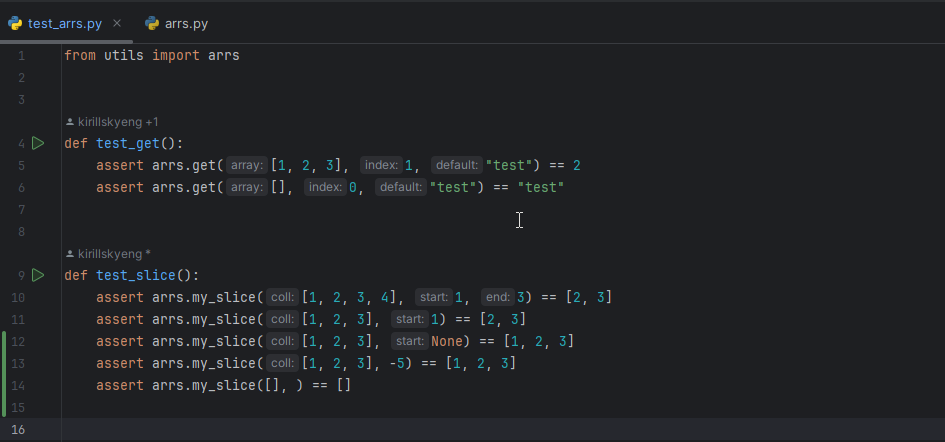


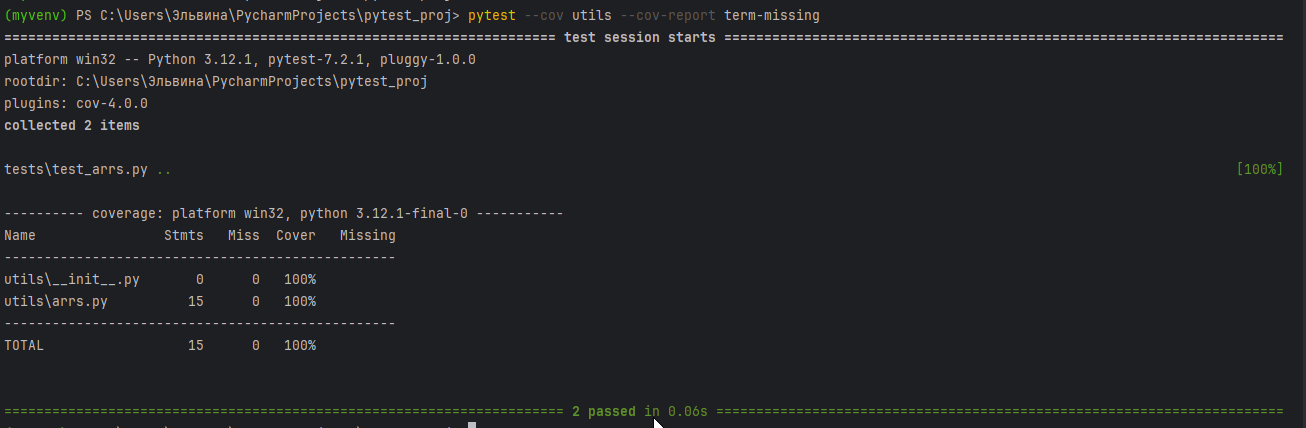


1. Допишите тесты на те функции, которые недостаточно покрыты тестами, опираясь на описание их работы и отчет по покрытию.

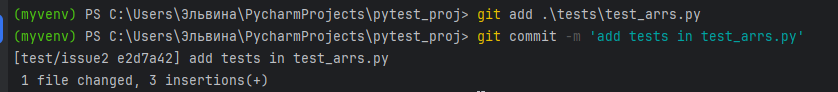


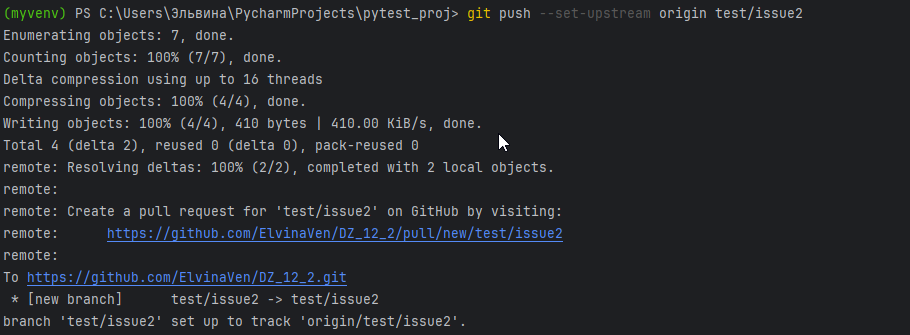
1. Добейтесь 100% покрытия тестами.

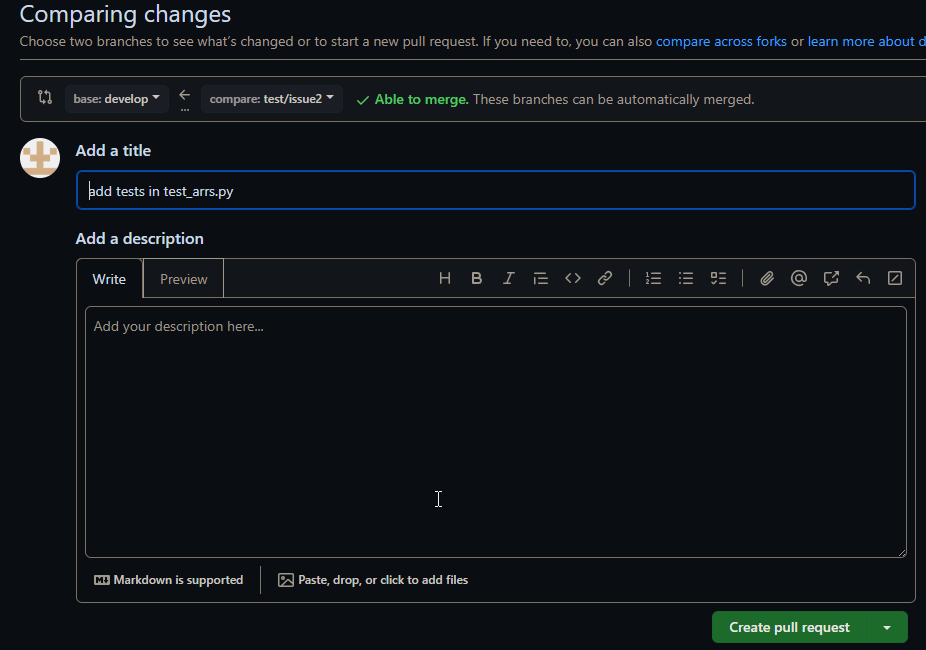


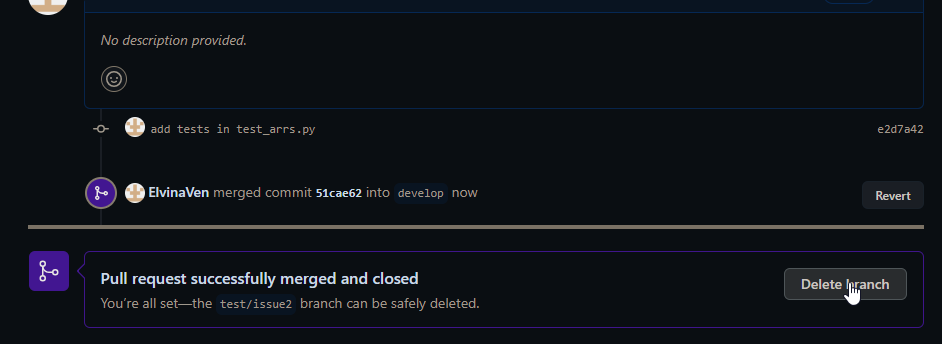


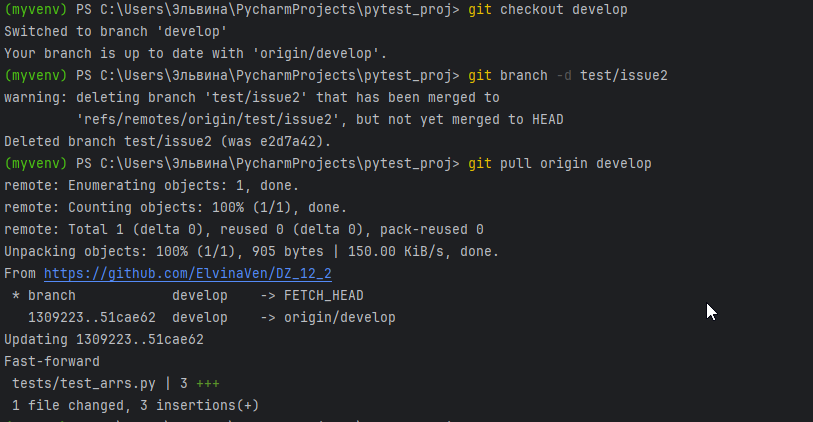
1. Сделайте pull request проделанной работы в ветку develop.







1. После апрува пул-реквеста удалите ветку test/issue2 (прямо на GitHub). 
2. Также удалите локальную ветку test/issue2 и обновите ветку develop.

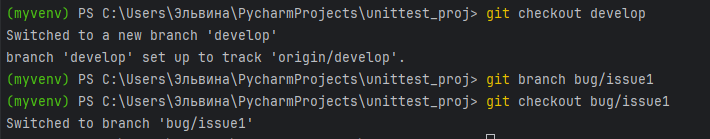


После всего в пайчарме git fetch и обновятся все ветки

Задания 3 и 4 дублируют задания 1 и 2. Но в следующих заданиях реализуйте тесты с **помощью библиотеки** unittest, а не фреймворка pytest.

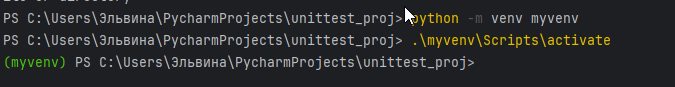
**Задание 3. Фикс тестов**

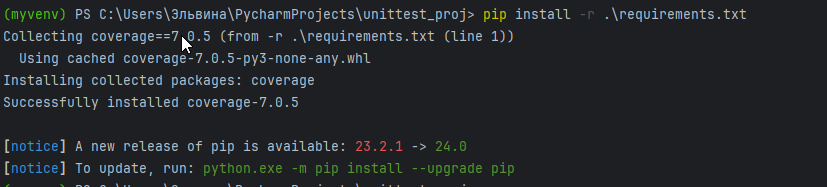
Работу по этой задаче ведите в ветке bug/issue1.



**Выполните следующие шаги:**

1. Клонируйте проект <https://github.com/kirillskyeng/unittest_proj>. Создайте виртуальное окружение, установите зависимости.





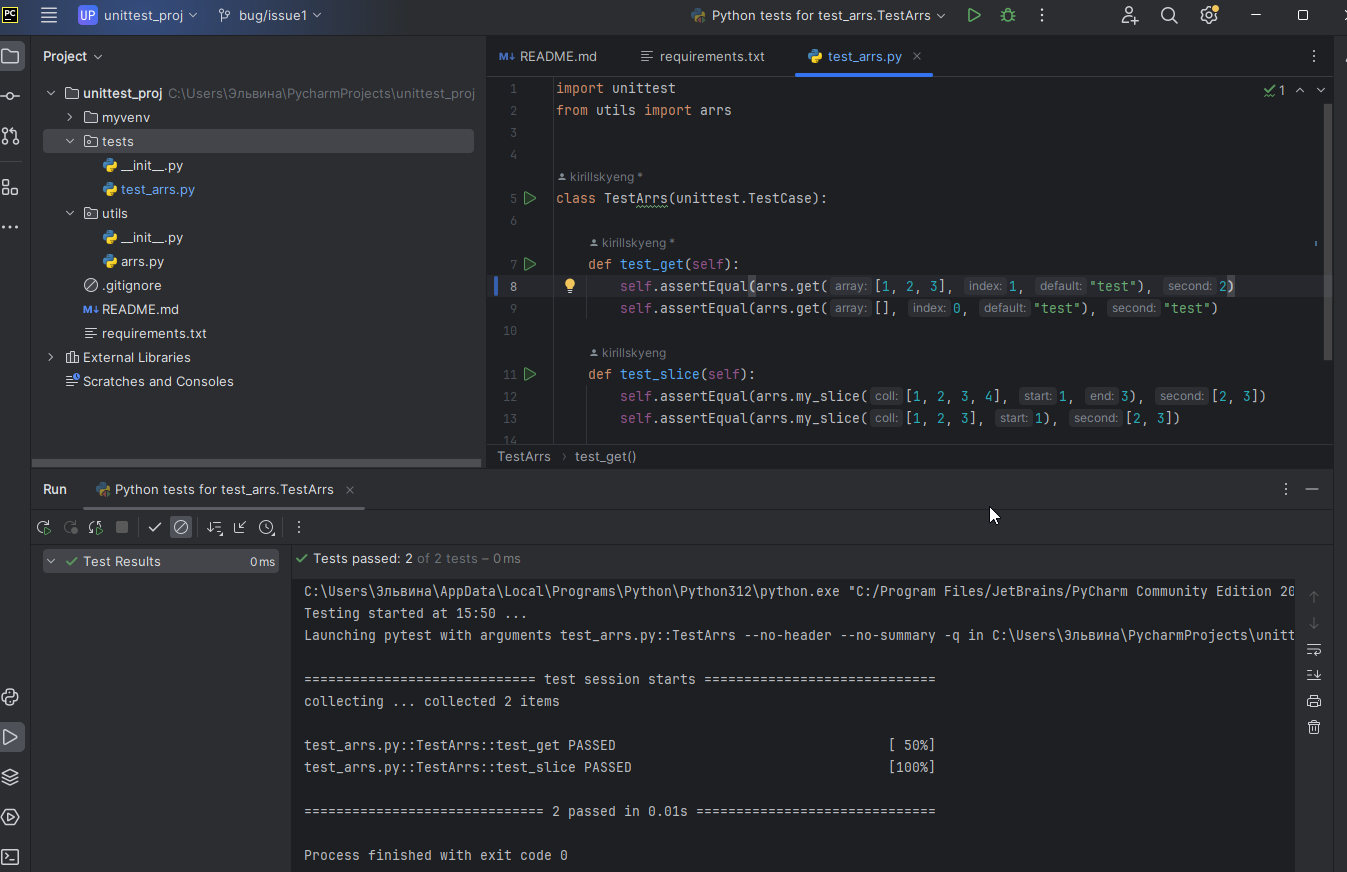
1. Разберитесь с его структурой. В проекте есть пакеты:
   * utils

 — пакет, содержащий модули для работы с массивами;

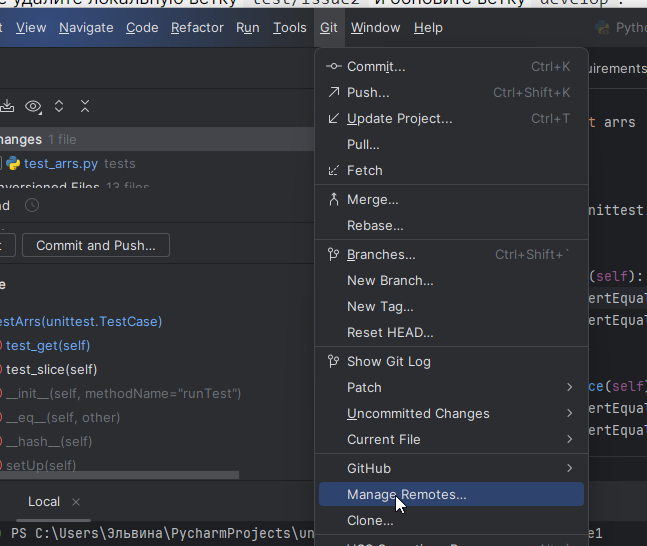
* + tests

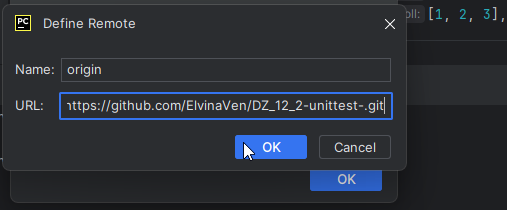
 — пакет с тестами.

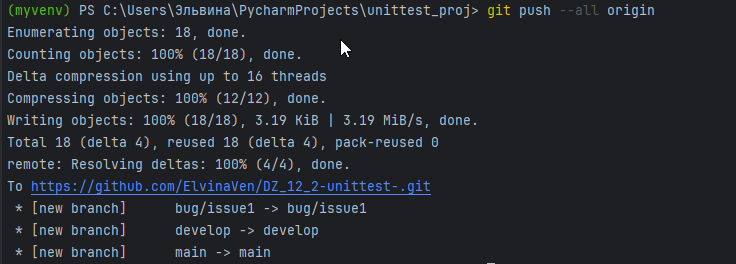
1. Добейтесь того, чтобы тесты, написанные к модулю arrs, проходили. Тесты пишите с помощью unittest.



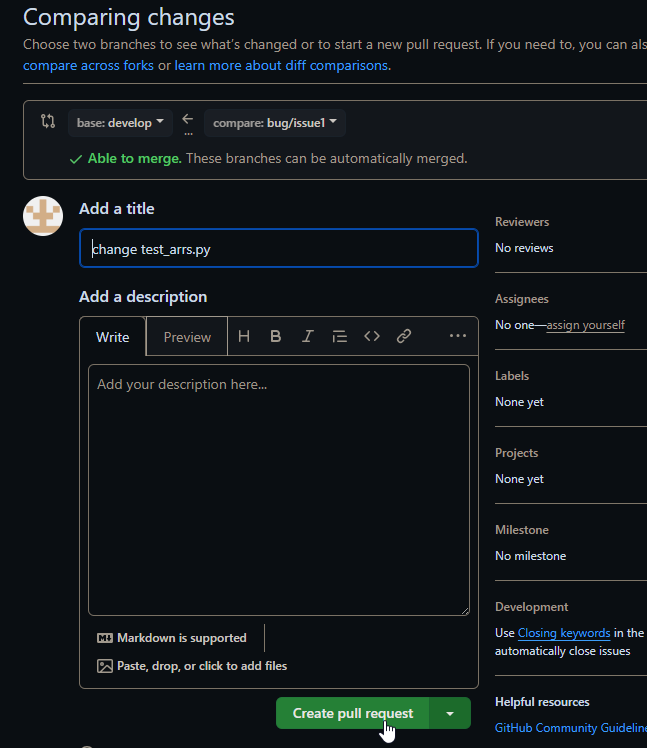
1. Залейте проект **себе** на GitHub, создав новый репозиторий.



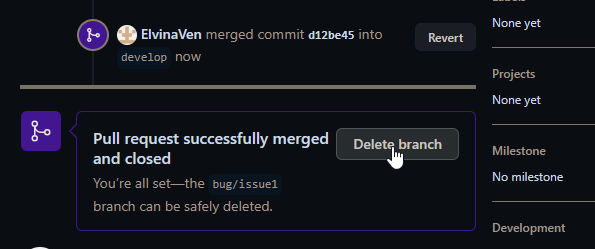




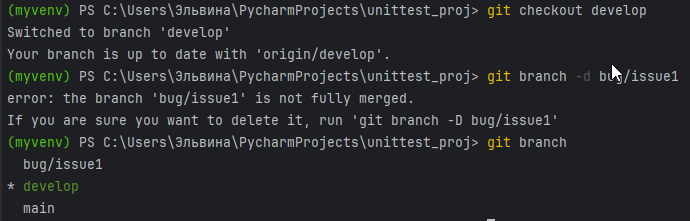
1. Сделайте pull request в ветку develop и за тимлида заапрувьте его.



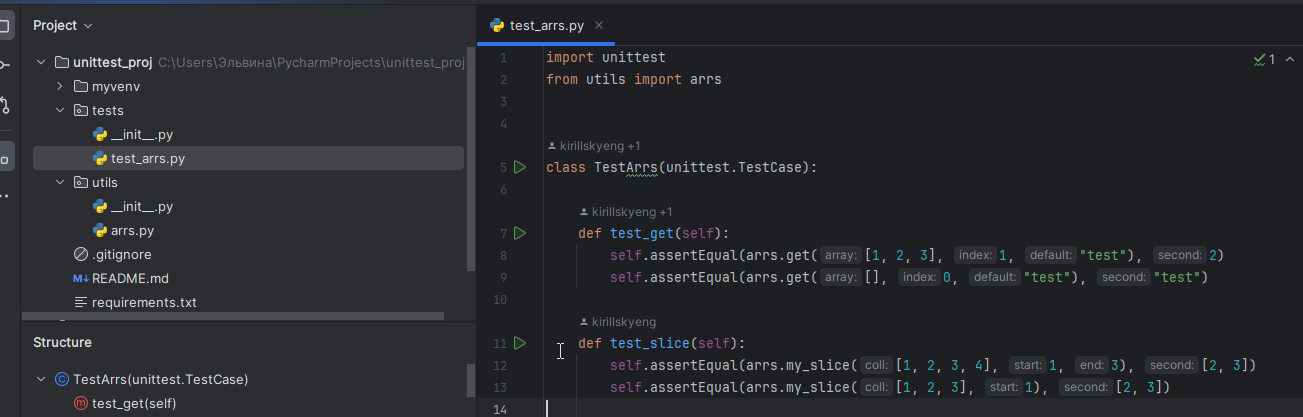
1. После апрува пул-реквеста удалите ветку bug/issue1 (прямо на GitHub).



1. Удалите локальную ветку bug/issue1 и обновите ветку develop.

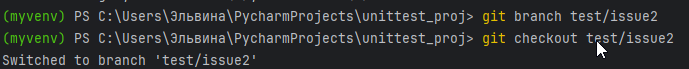






**Задание 4. Покрытие тестами**

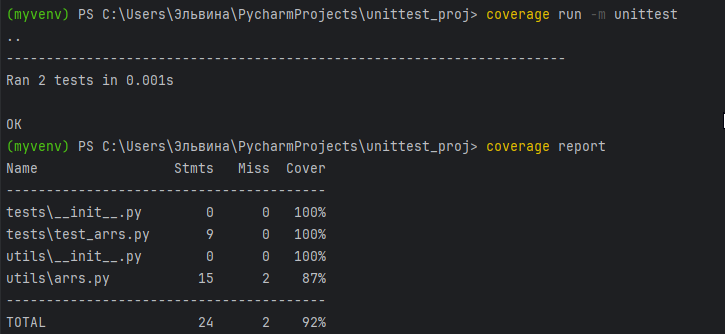
Работу по этой задаче ведите в ветке test/issue2.



**Выполните следующие шаги:**

1. Посмотрите на процент покрытия тестами в проекте.

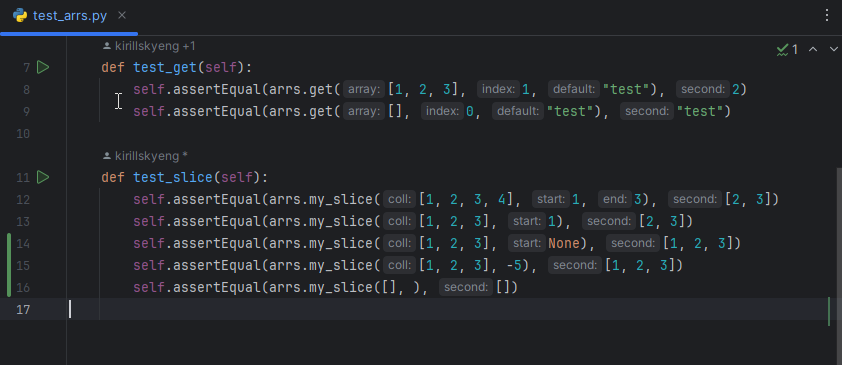
Чтобы запустить покрытие тестами, выполните в консоли, находясь в виртуальном окружении проекта, команду coverage run -m unittest для создания отчета, а далее команду coverage report для отображения.



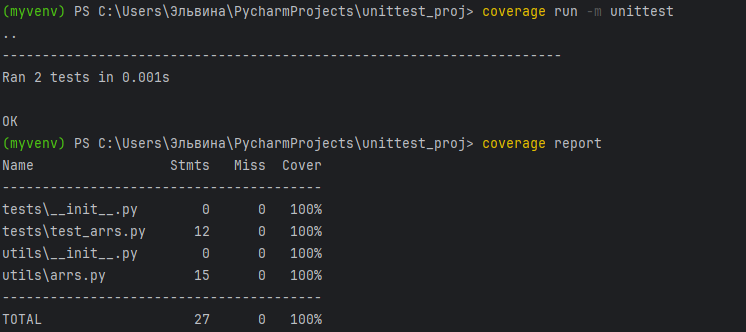
Или одной командой: coverage run -m unittest && coverage report

Также можно сгенерировать HTML-отчет с помощью команды:python -m coverage html

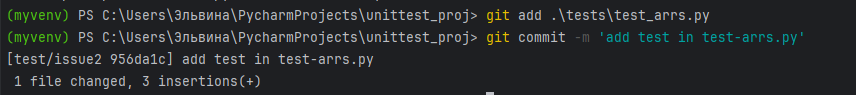
1. Допишите тесты на те функции, которые недостаточно покрыты тестами, опираясь на описание их работы и отчет по покрытию.

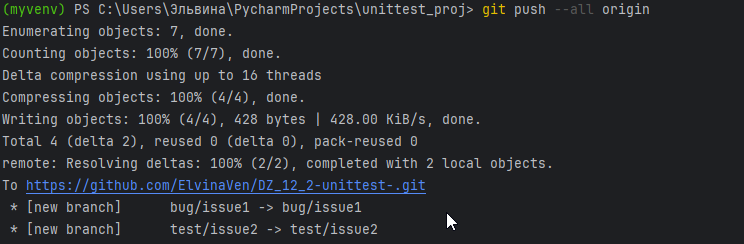


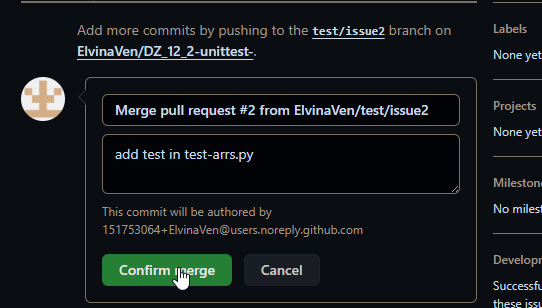
1. Добейтесь 100% покрытия тестами.

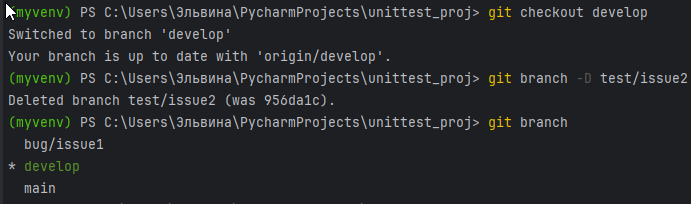


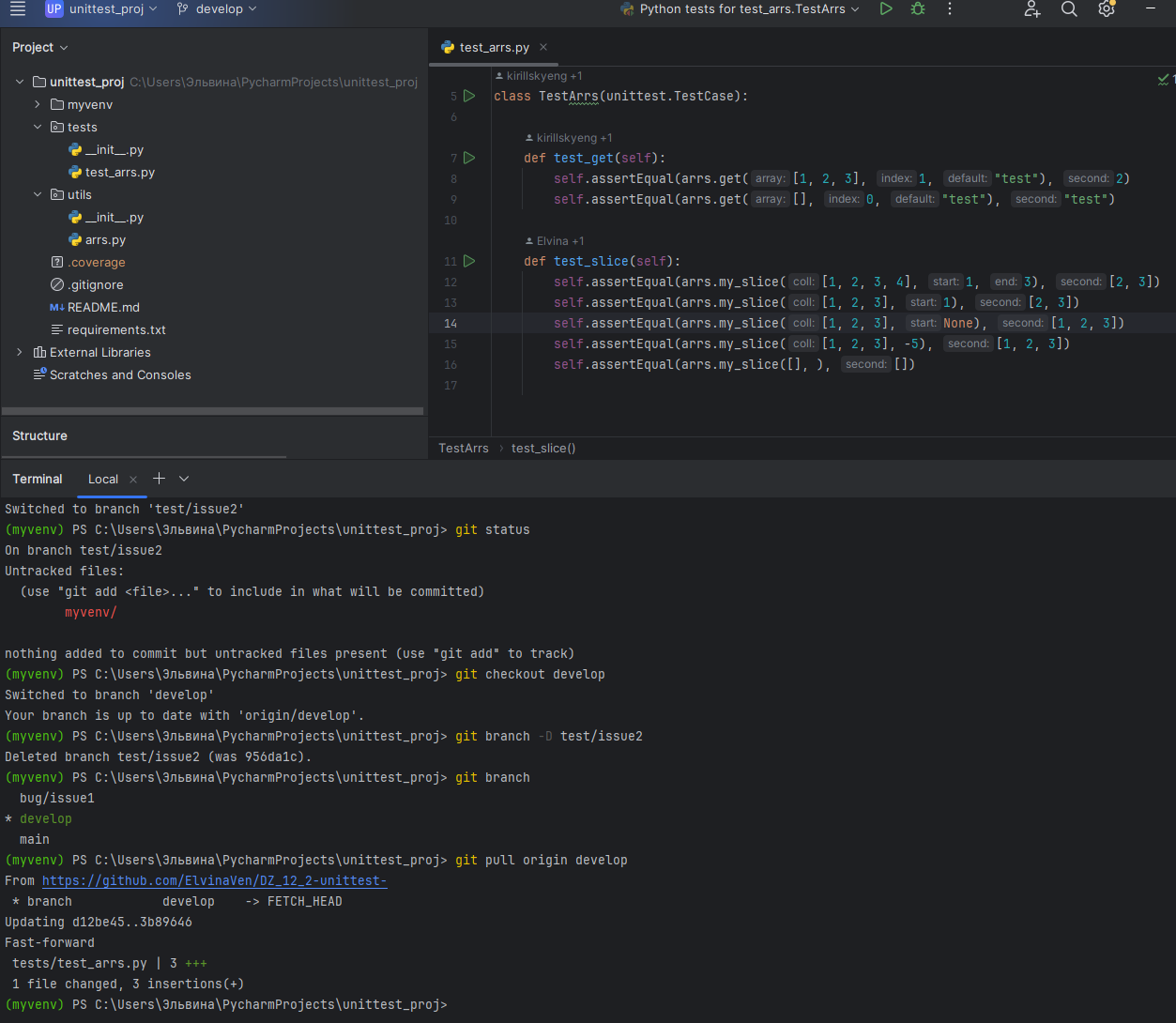
1. Сделайте pull request проделанной работы в ветку develop.











**Задание 5. Функция get\_val**

Вы разобрались с существующим кодом в проекте и начинаете реализацию новой функциональности. Сделайте модуль, который содержит функции для работы со словарями, и пропишите к ним тесты.

Попробуйте реализовать подход TDD (Test Driven Development) и сначала написать тесты к функциям, а потом сами функции.

Работу над этой задачей реализуйте либо с pytest, либо с unittest.

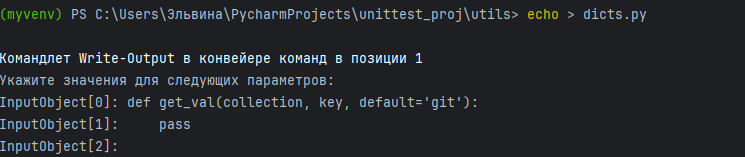
**Выполните следующие шаги:**

1. Создайте в utils/ файл dicts.py и поставьте в нем заглушку на будущую реализацию функции:

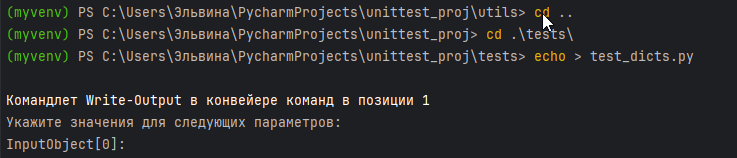
# файл dicts.py

def get\_val(collection, key, default='git'):

pass



1. Напишите минимум три теста для функции get\_val, исходя из описания ее функционала.



**Описание будущего функционала функции**

Функция возвращает значение из словаря по переданному ключу, если ключ существует. В ином случае возвращается значение default.

Примеры работы:

>>> data = {"vcs": "mercurial"}

>>> get\_val(data, "vcs")

'mercurial'

>>> get\_val(data, "vcs", "git")

'mercurial'

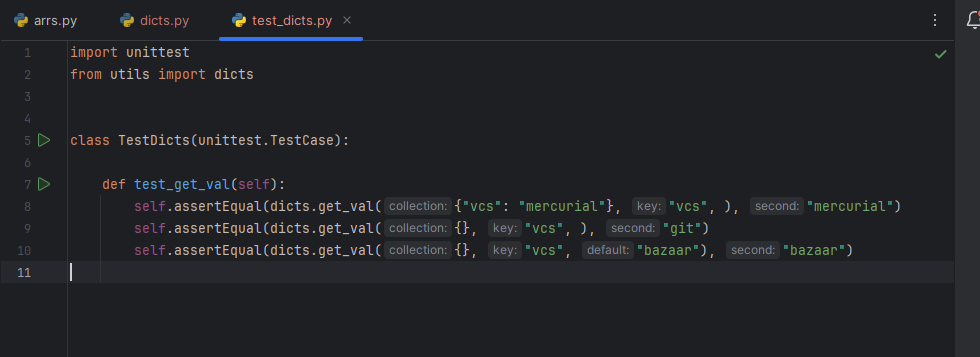
>>> data = {}

>>> get\_val({}, "vcs", "git")

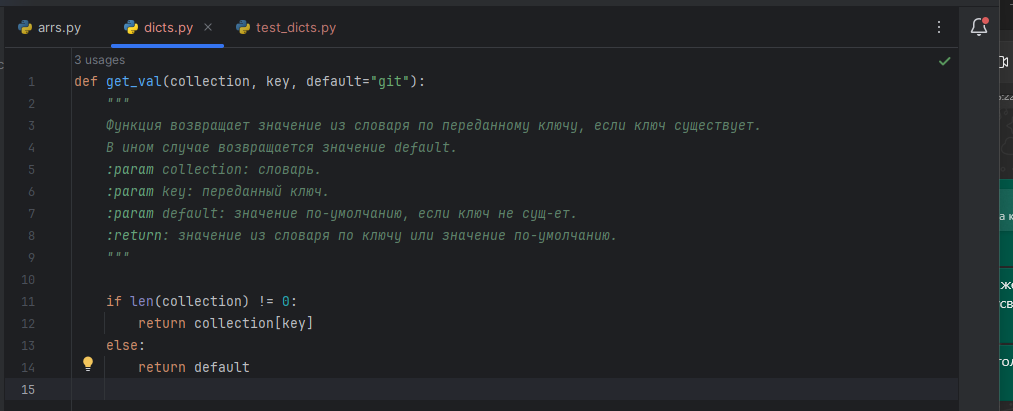
'git'

>>> get\_val({}, "vcs", "bazaar")

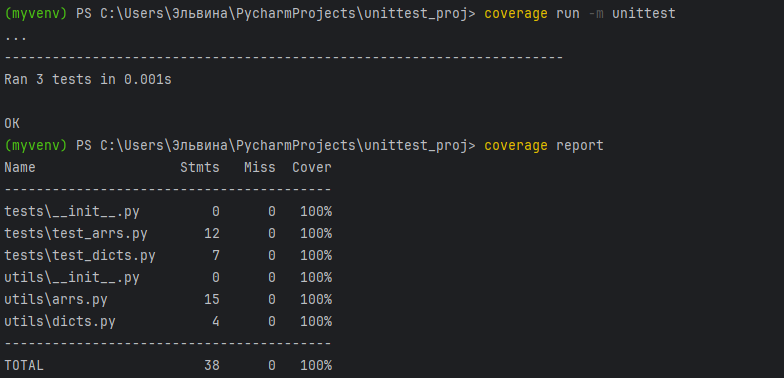
'bazaar'



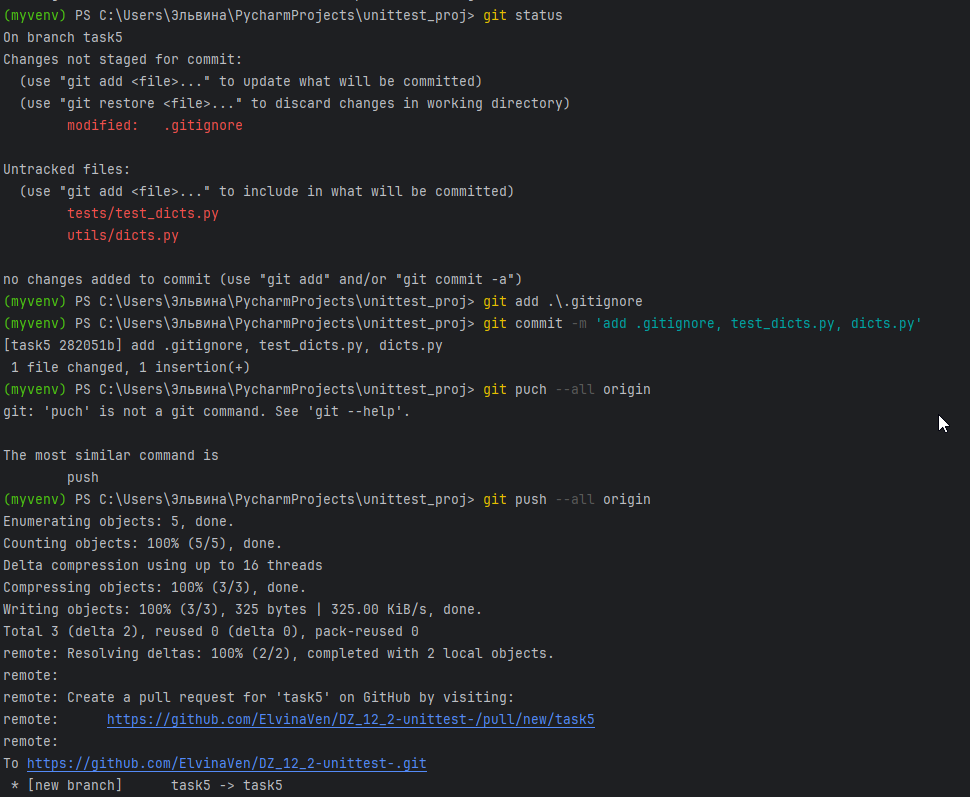
1. Напишите реализацию функции. По ходу разработки запускайте тесты, чтобы смотреть, какие кейсы еще остались незакрытыми.

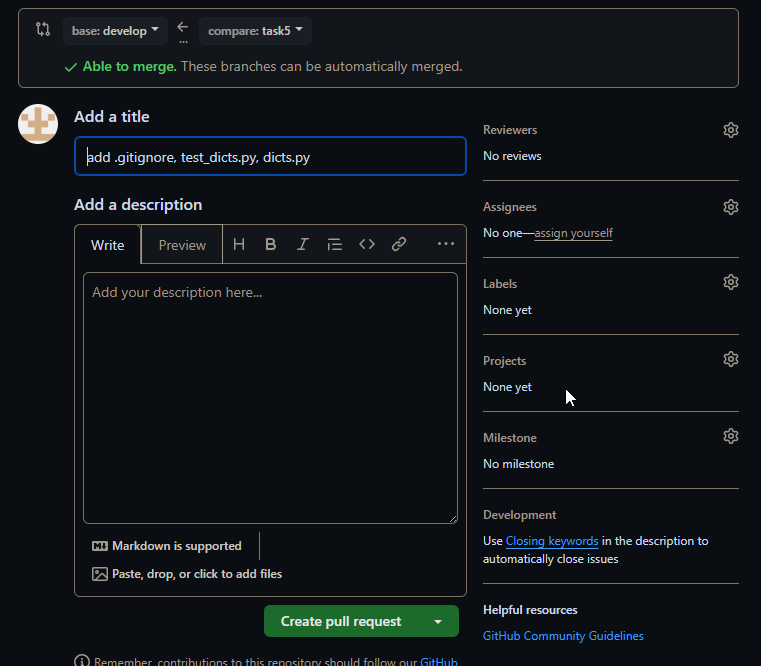


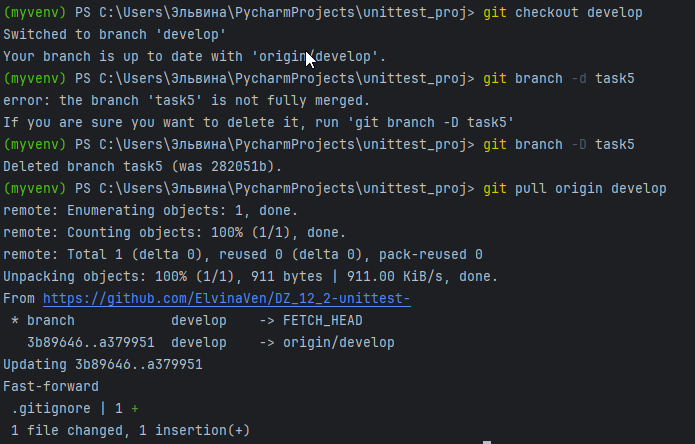
1. Запустите тесты, убедитесь, что они все проходят.

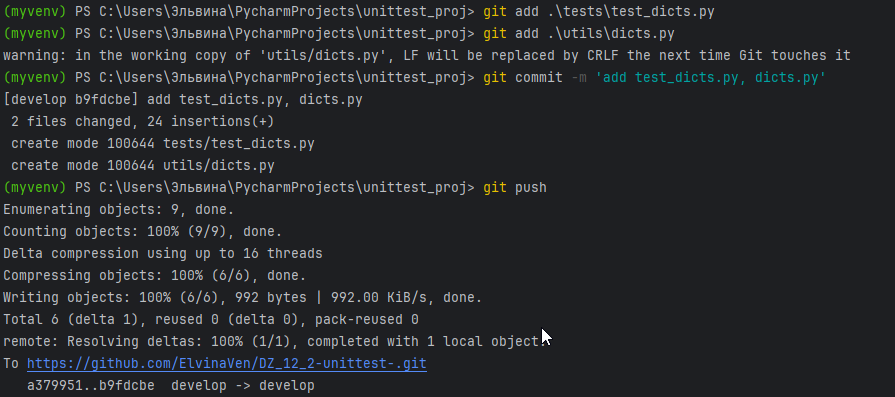


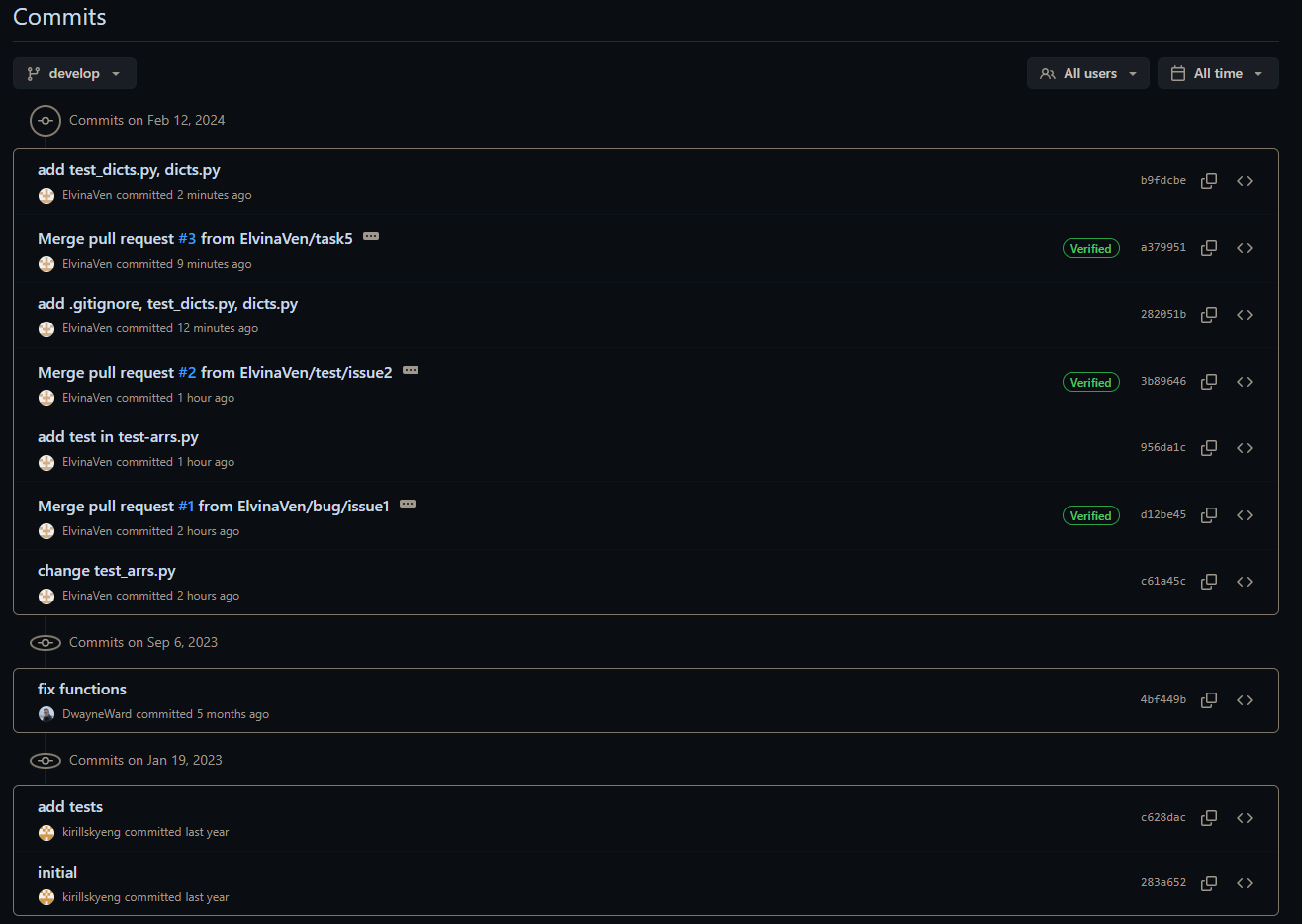
1. Сделайте pull request.











**Задание 6. Фикстуры**

Улучшите тесты из задания 2, используйте для этого фикстуры pytest.



