

Введение в многозадачность

Семинар 4





Что будет сегодня на семинаре. Наши цели:

- Узнать о многопоточном подходе в программировании
- Разобраться с многопроцессорным подходом в Python
- Изучить асинхронный подход в решении задач



Вопросы?

Вопросы?



Вопросы?





Задание №1

- Написать программу, которая считывает список из 10 URL-адресов и одновременно загружает данные с каждого адреса.
- После загрузки данных нужно записать их в отдельные файлы.
- Используйте потоки.

The screenshot shows a Django project in VS Code. The main editor displays a test file `test_index_view.py` with a failing test `test_index_view_with_a_future_question`. The test failure message is: `AssertionError: [b'None'] != [b'Past question.'].', {'question': Past question.}].` The test code is as follows:

```
def test_index_view_with_a_future_question(self):
    """
    Questions with a pub date in the future show
    the index page.
    """
    create_question(question_text="Future question.", days=30)
    response = self.client.get(reverse("polls:index"))
    self.assertContains(response, "No polls are available.")
    self.assertQuerysetEqual(response.context['latest_question_list'],
        [{"question": Past question.}])

def test_index_view_with_future_question_and_past_question(self):
    """
    Even if both past and future questions exist
    only the past question is displayed.
    """
    create_question(question_text="Past question.", days=-30)
    create_question(question_text="Future question.", days=30)
    response = self.client.get(reverse("polls:index"))
    self.assertQuerysetEqual(
        response.context['latest_question_list'],
        [{"question": Past question.}])
```

The Django ORM log on the right shows the following queries:

```
SELECT "polls_question"."id", "polls_question"."question_text", "polls_question"."pub_date" FROM "polls_question" WHERE "polls_question"."pub_date" < 2018-10-10 11:35:52.145234;
SELECT "polls_question"."id", "polls_question"."question_text", "polls_question"."pub_date" FROM "polls_question" WHERE "polls_question"."pub_date" > 2018-10-10 11:35:52.145234;
```

The bottom panel shows the Django ORM log with the following entries:

```
DEBUG:django.db.backends:SELECT "polls_question"."id", "polls_question"."question_text", "polls_question"."pub_date" FROM "polls_question" WHERE "polls_question"."pub_date" < 2018-10-10 11:35:52.145234;
DEBUG:django.db.backends:SELECT "polls_question"."id", "polls_question"."question_text", "polls_question"."pub_date" FROM "polls_question" WHERE "polls_question"."pub_date" > 2018-10-10 11:35:52.145234;
```

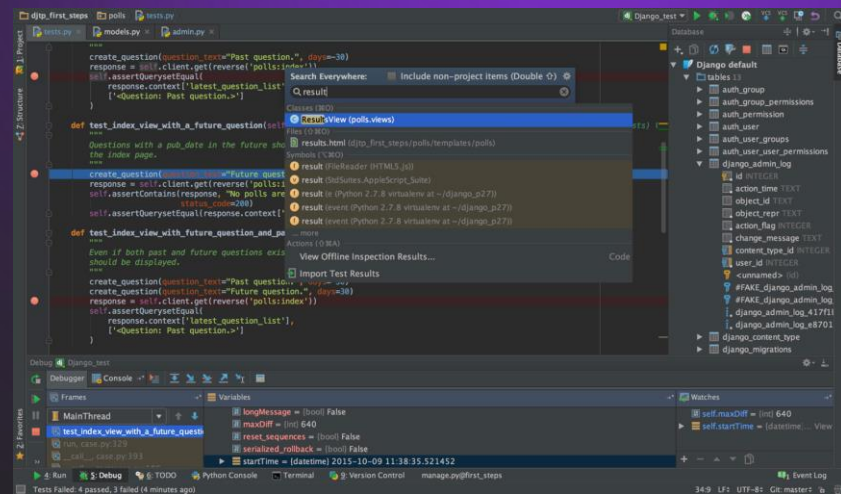


Задание №2

- Написать программу, которая считывает список из 10 URL-адресов и одновременно загружает данные с каждого адреса.
- После загрузки данных нужно записать их в отдельные файлы.
- Используйте процессы.



Решение в IDE



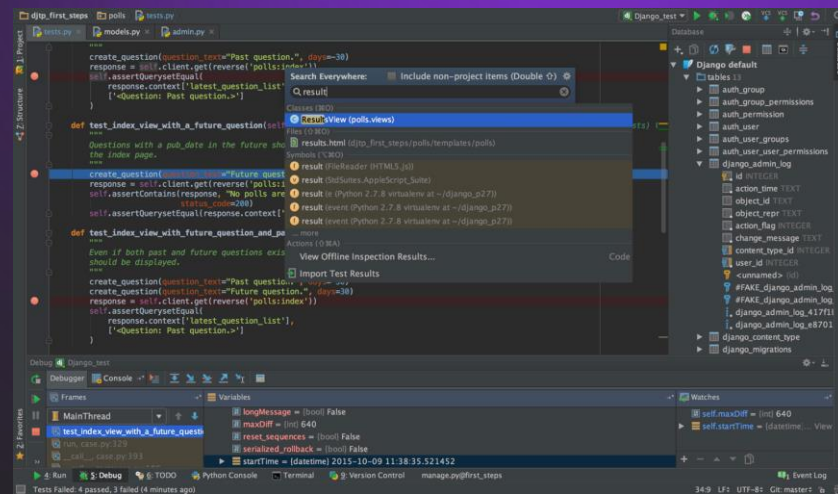


Задание №3

- Написать программу, которая считывает список из 10 URL-адресов и одновременно загружает данные с каждого адреса.
- После загрузки данных нужно записать их в отдельные файлы.
- Используйте асинхронный подход.



Решение в IDE



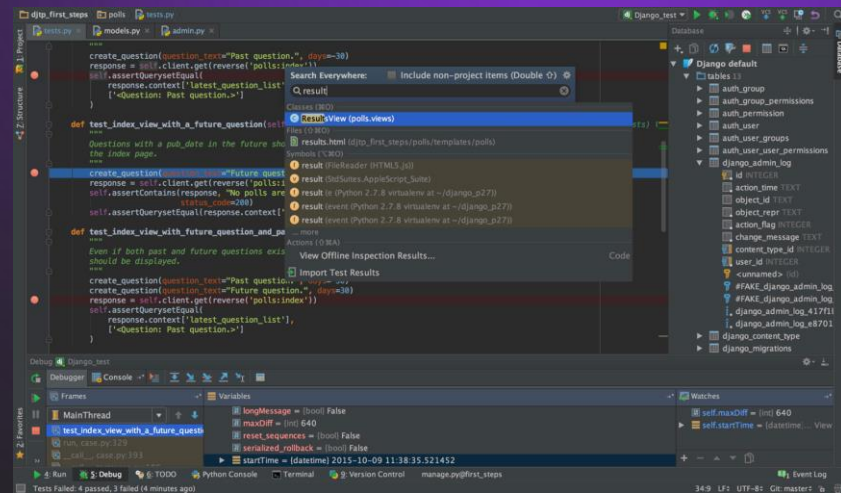


Задание №4

- Создать программу, которая будет производить подсчет количества слов в каждом файле в указанной директории и выводить результаты в консоль.
- Используйте потоки.



Решение в IDE



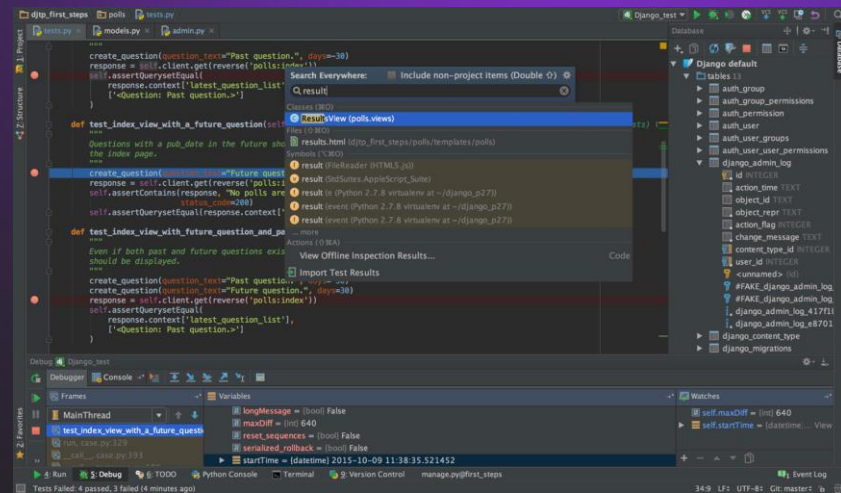


Задание №5

- Создать программу, которая будет производить подсчет количества слов в каждом файле в указанной директории и выводить результаты в консоль.
- Используйте процессы.



Решение в IDE



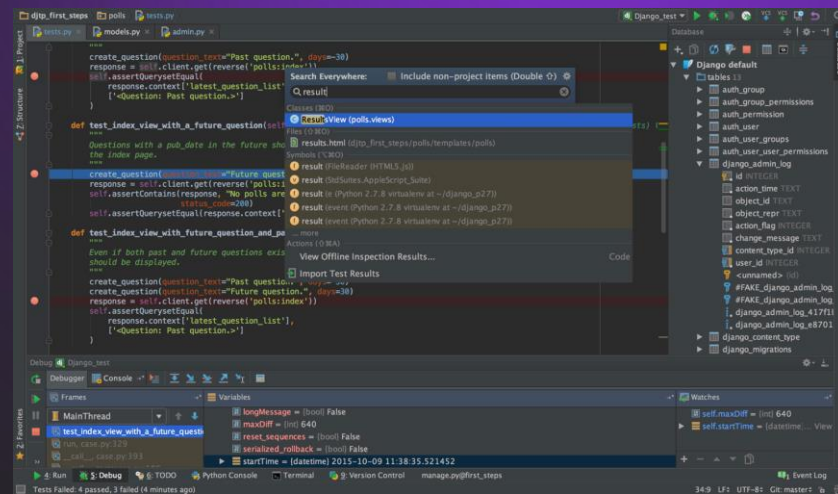


Задание №6

- Создать программу, которая будет производить подсчет количества слов в каждом файле в указанной директории и выводить результаты в консоль.
- Используйте асинхронный подход.



Решение в IDE





Перерыв?

Голосуйте в чате



Перерыв...

<<7:00->>

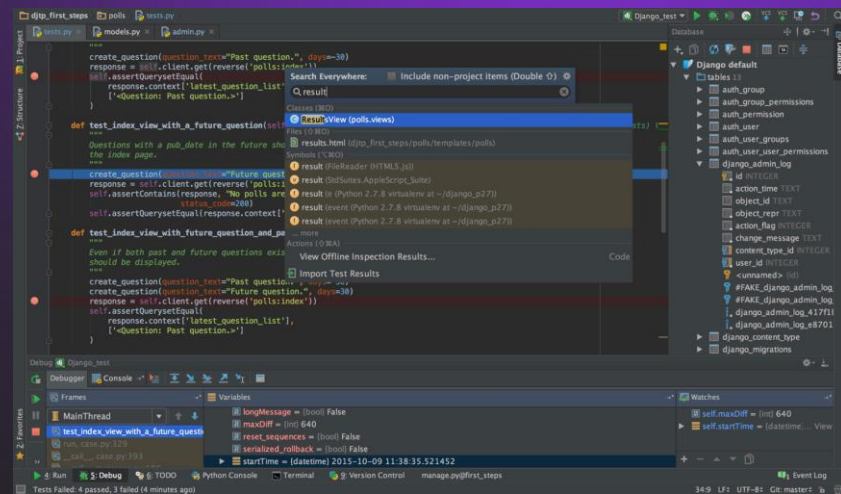


Задание №7

- Напишите программу на Python, которая будет находить сумму элементов массива из 1000000 целых чисел.
- Пример массива: `arr = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ...]`
- Массив должен быть заполнен случайными целыми числами от 1 до 100.
- При решении задачи нужно использовать многопоточность, многопроцессорность и асинхронность.
- В каждом решении нужно вывести время выполнения вычислений.



Решение в IDE



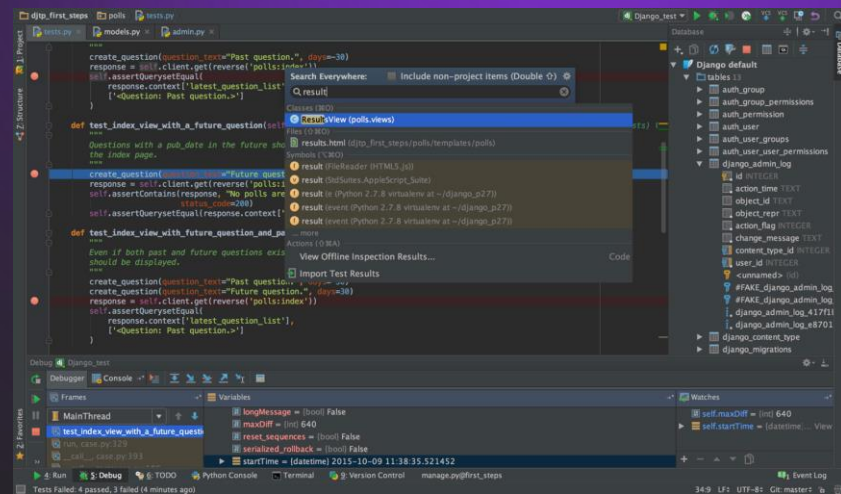


Задание №8

- Напишите программу, которая будет скачивать страницы из списка URL-адресов и сохранять их в отдельные файлы на диске.
- В списке может быть несколько сотен URL-адресов.
- При решении задачи нужно использовать многопоточность, многопроцессорность и асинхронность.
- Представьте три варианта решения.



Решение в IDE

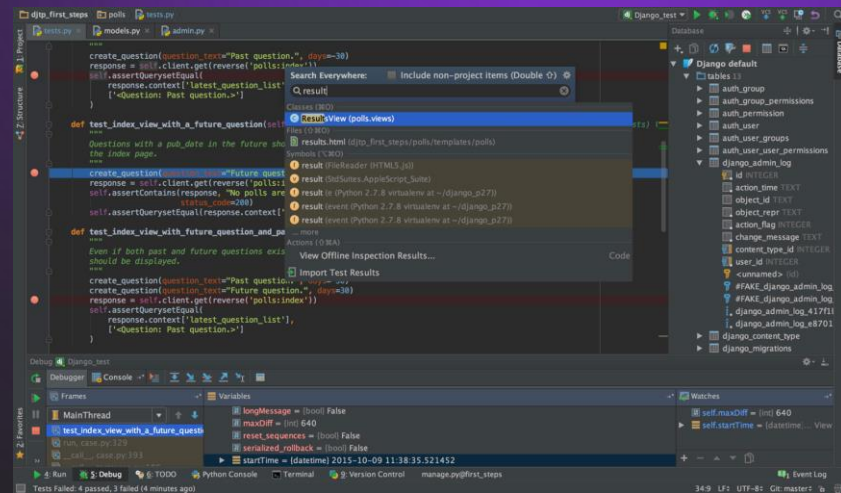




Задание №9

- Написать программу, которая скачивает изображения с заданных URL-адресов и сохраняет их на диск. Каждое изображение должно сохраняться в отдельном файле, название которого соответствует названию изображения в URL-адресе.
- Например URL-адрес: `https://example/images/image1.jpg` -> файл на диске: `image1.jpg`
- Программа должна использовать многопоточный, многопроцессорный и асинхронный подходы.
- Программа должна иметь возможность задавать список URL-адресов через аргументы командной строки.
- Программа должна выводить в консоль информацию о времени скачивания каждого изображения и общем времени выполнения программы.

Решение в IDE





Вопросы?

Вопросы?



Вопросы?





Домашнее задание



Задание

- Решить задачи, которые не успели решить на семинаре.



Подведем итоги



Что было
сложного на
семинаре?





Напишите три вещи в
комментариях, которым
вы научились сегодня.





Как настроение?





Спасибо за работу!