**2023/10/22 00:00|Лекция. Практика. Часть 2.**

**Практика. Часть 2**

На этом занятии продолжаем работать с нашим пакетом. По итогу добавили ещё 1 алгоритм сортировки, чтобы у нас файл не был слишком пустым. Получилось нечто, напоминающее модуль, содержащий в себе несколько функций, которые могут использовать ваши коллеги(рис.1).**Как это можно применить?**

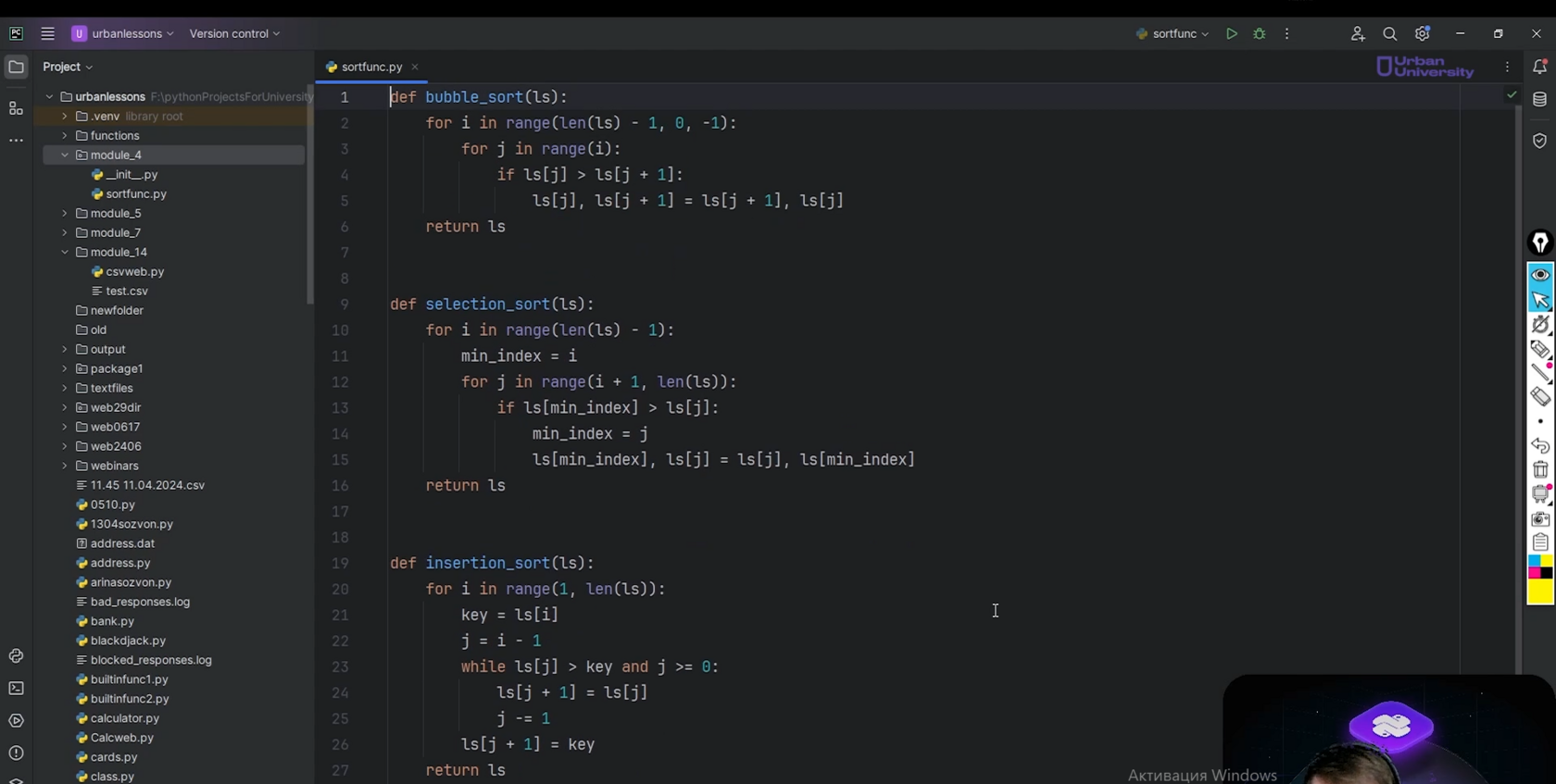


Рис.1

Вот у вас есть пакет и представим, что кто-то у вас работает. Вы поделились вашим пакетом модуль 4 (просто название). Он содержит в себе модуль sortfunc. Внутри находятся функции для сортировки, которые вы можете использовать. Ребята пишут какую-нибудь программу, какой-нибудь проект, например, arrays для работы со списками и имеют списки разной длины, с любыми данными(рис.2):

data\_1 = [9, 7, 4, 5, 6, 7, 1, 2]

data\_2 = [9, 7, 4, 5, 6, 7, 1, 2, 10, 1]

data\_3 = [9, 7, 4, 5, 6, 7, 1, 2, 1, 2, 3]

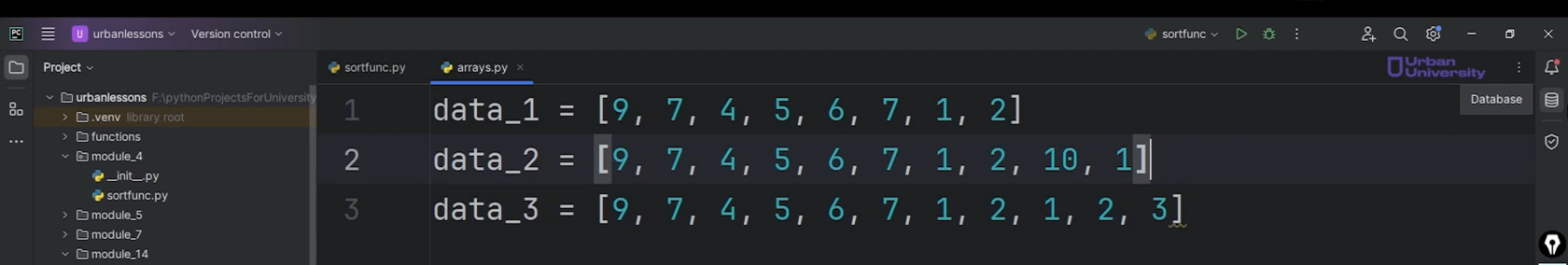


Рис.2

Суть заключается в том, что ребята делают проект, который берет списки, вычисляет наиболее оптимальный способ для сортировки этих списков или проводят какой-то анализ “Как от количества элементов меняется скорость использования той или иной сортировки?”. Для этой задачи они будут использовать ваш пакет, то есть им всего лишь то остаётся взять из вот этого пакета модуль 4, модуль sortfunc, импортировать функции для сортировки insertion\_sort, bubble\_sort и selection\_sort(рис.3):

From module\_1.sortfunc import insertion\_sort, bubble\_sort, selection\_sort

**Здесь, в принципе, можно сделать так:**

print(insertion\_sort(data\_1))

print(insertion\_sort(data\_2))

print(insertion\_sort(data\_3))

print(bubble\_sort(data\_1))

print(bubble\_sort(data\_2))

print(bubble\_sort(data\_3))

print(selection\_sort(data\_1))

print(selection\_sort(data\_2))

print(selection\_sort(data\_3))

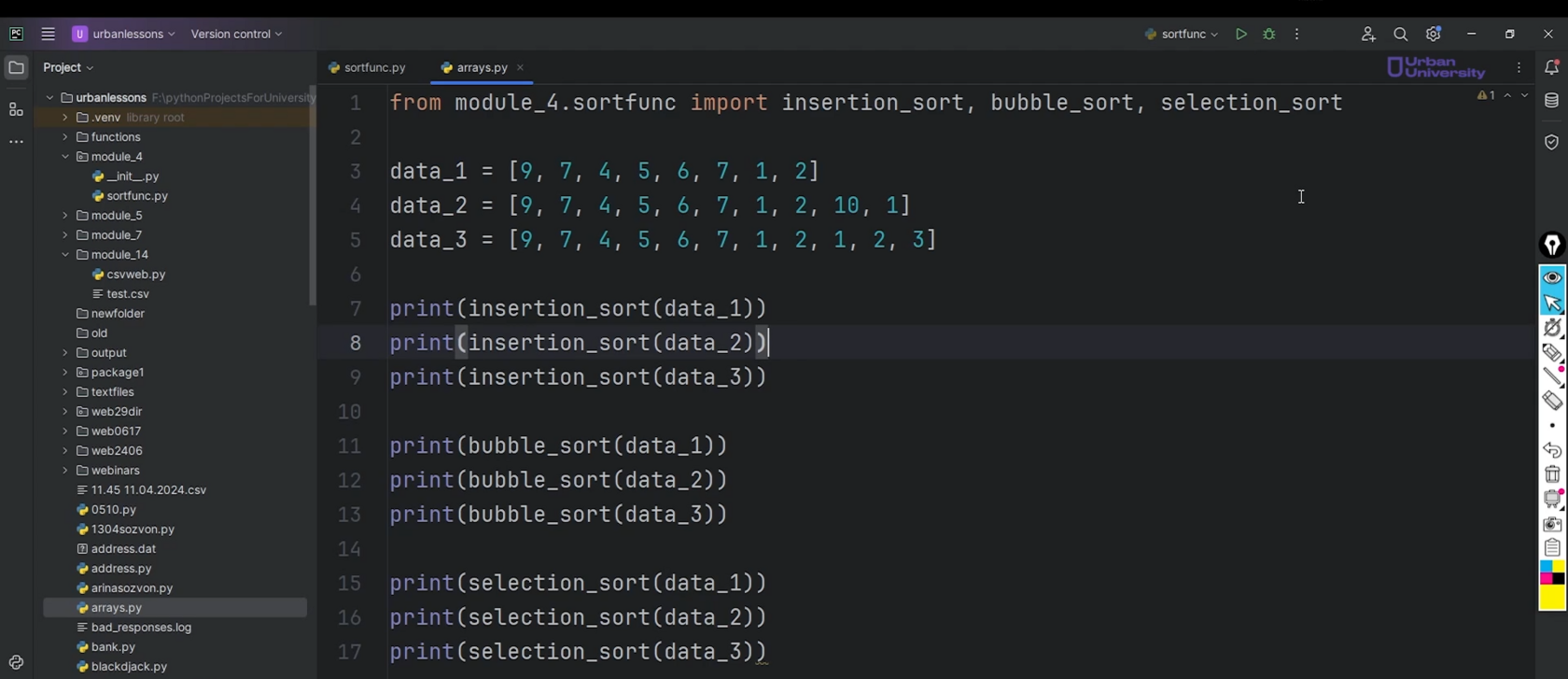


Рис.3

**То есть будем выводить результат работы каждой функции каждого способа сортировки, передавая туда все эти массивы, все эти списки с данными, но разных размеров.**

Для самостоятельной практики вы можете попробовать обернуть эти функции или добавить внутрь этого пакета модуль с функционалом, который будет замерять время выполнения функции и применить его здесь. Это поможет пользователю, при запуске программы, наглядно видеть скорость выполнения. Также пользователь сможет сравнить её, чтобы понять, какой из этих алгоритмов выполняется быстрее. Сейчас мы просто пытаемся воспроизвести ситуацию, когда полезно использовать вот такие настройки в вашей программе.

По итогу после запуска, в принципе, мы видим, что каждая из этих функций успешно справлялась со своей задачей(рис.4). Все работают корректно. По сути, вы сделали некую надстройку, вынесли туда определённый функционал и позволили его использовать вам и вашим коллегам без необходимости придумывать что-то и засорять основной код.

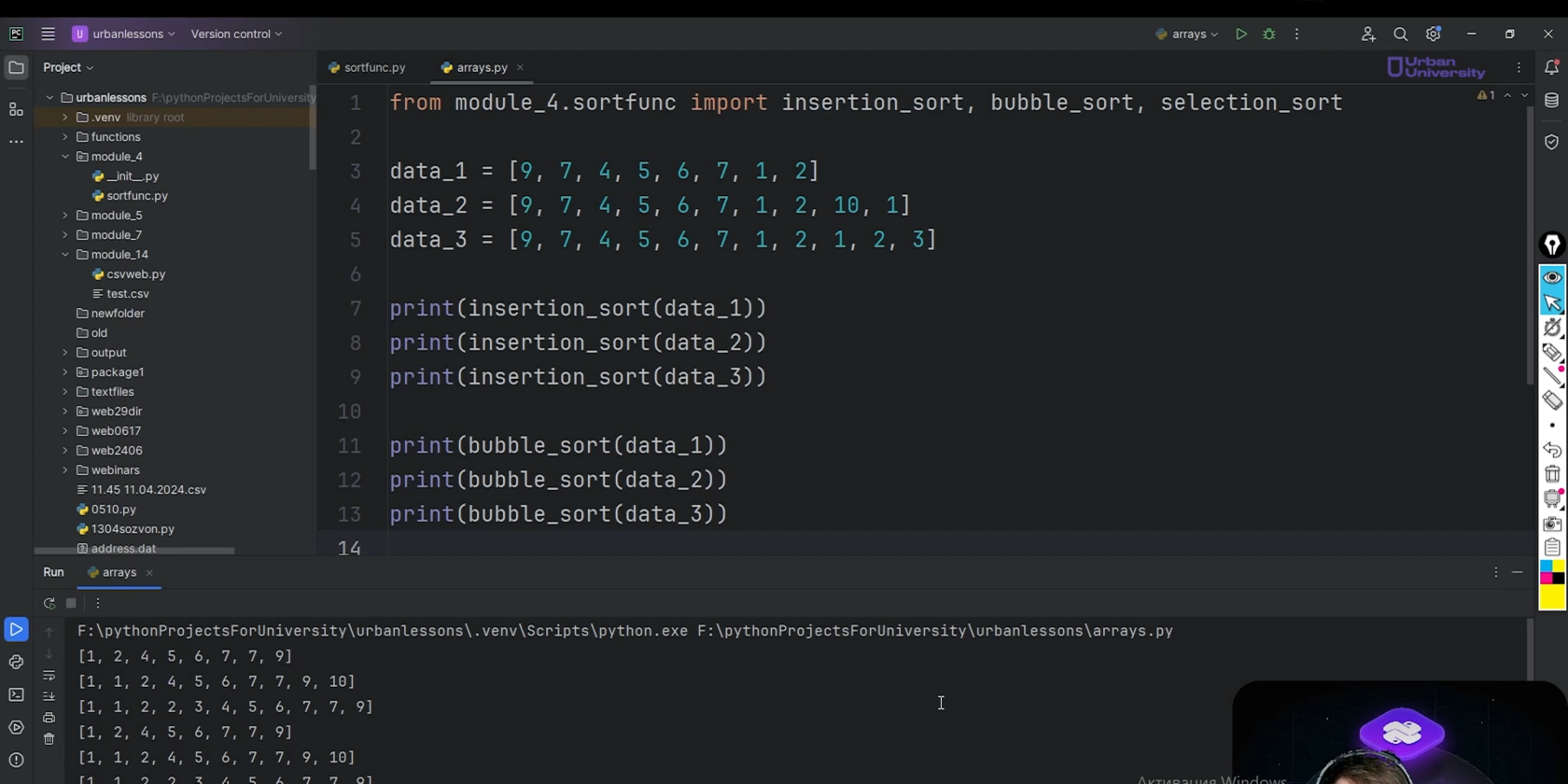


Рис.4