

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

**«Лабораторная работа 2.14 Установка
пакетов в Python. Виртуальные
окружения»**

**ОТЧЕТ
по лабораторной работе №17
дисциплины
«Основы программной инженерии»**

Выполнил:
Луценко Дмитрий Андреевич
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

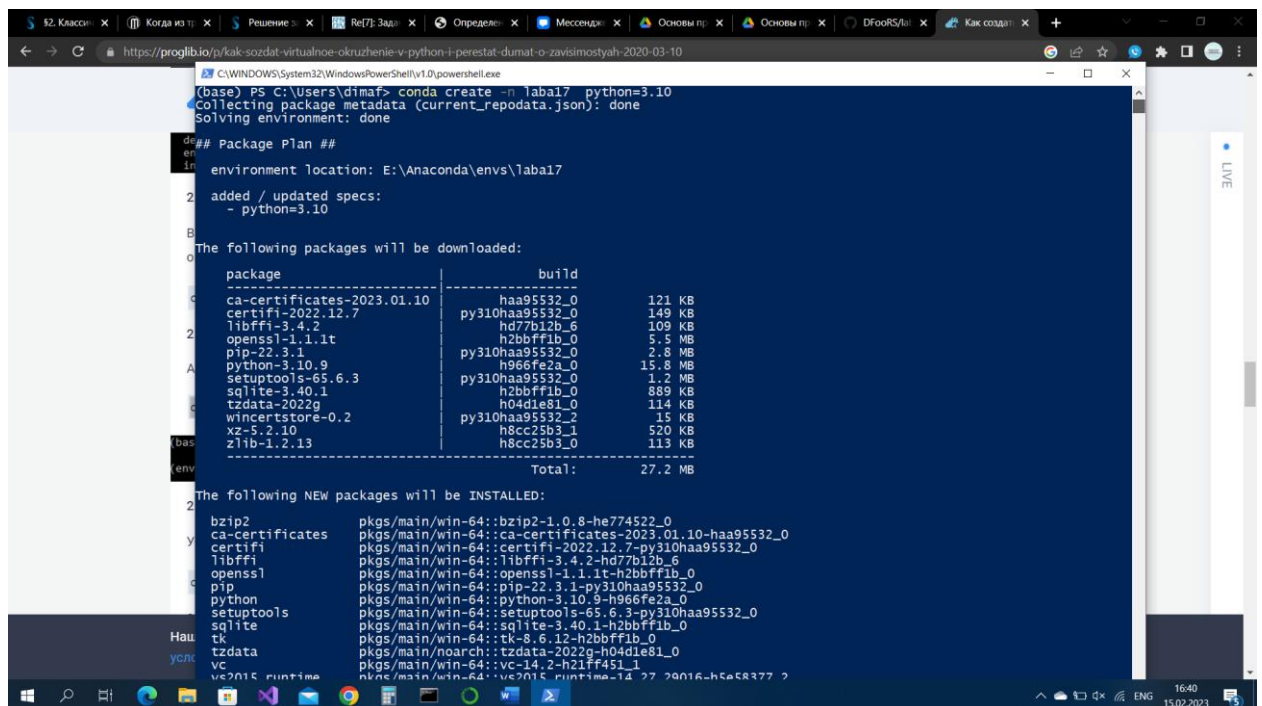
Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Лабораторная работа 2.14 Установка пакетов в Python. Виртуальные окружения

Цель работы: приобретение навыков по работе с менеджером пакетов `pip` и виртуальными окружениями с помощью языка программирования Python версии 3.x.

Ход работы:



```
(base) PS C:\Users\dima\> conda create --name lab17 python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##
  environment location: E:\Anaconda\envs\lab17
  added / updated specs:
    - python=3.10

The following packages will be downloaded:

package                                build                                121 KB
ca-certificates-2023.01.10             haa95532_0                          149 KB
certifi-2022.12.7                      py310haa95532_0                     109 KB
libffi-3.4.2                           hd77b12b_6                          5.5 MB
openssl-1.1.1t                         py310haa95532_0                     2.8 MB
python-3.10.9                          h966fe2a_0                          15.8 MB
setuptools-65.6.3                      py310haa95532_0                     1.2 MB
sqlite-3.40.1                          h2bbff1b_0                          889 KB
tzdata-2022g                           h04d1e81_0                          114 KB
wincertstore-0.2                       py310haa95532_2                     15 KB
xz-5.2.10                              h8cc25b3_1                          520 KB
zlib-1.2.13                            h8cc25b3_0                          113 KB
Total: 27.2 MB

The following NEW packages will be installed:

bz2pkgs/main/win-64::bz2pkgs/main/win-64::ca-certificates-2023.01.10-haa95532_0
certifipkgs/main/win-64::certifi-2022.12.7-py310haa95532_0
libffipkgs/main/win-64::libffi-3.4.2-hd77b12b_6
opensslpkgs/main/win-64::openssl-1.1.1t-h2bbff1b_0
pippkgs/main/win-64::pip-22.3.1-py310haa95532_0
pythonpkgs/main/win-64::python-3.10.9-h966fe2a_0
setuptoolspkgs/main/win-64::setuptools-65.6.3-py310haa95532_0
sqlitepkgs/main/win-64::sqlite-3.40.1-h2bbff1b_0
tkpkgs/main/win-64::tk-8.6.12-h2bbff1b_0
tzdatapkgs/main/noarch::tzdata-2022g-h04d1e81_0
vcpkgs/main/win-64::vc-14.2-h21ff451_1
vc2015runtimepkgs/main/win-64::vc2015runtime-14.27.26016-h5a6c8277_2
```

Рисунок 1 – Создание виртуального окружения

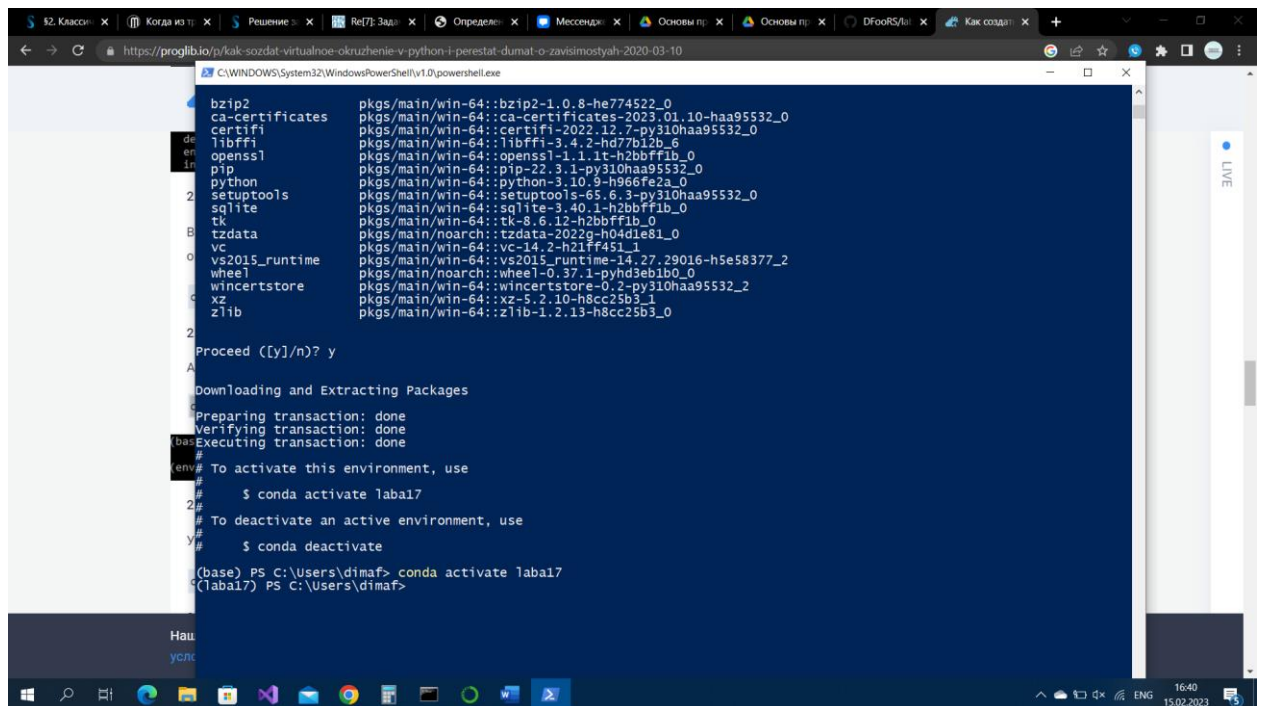


Рисунок 2 – Активация виртуального окружения

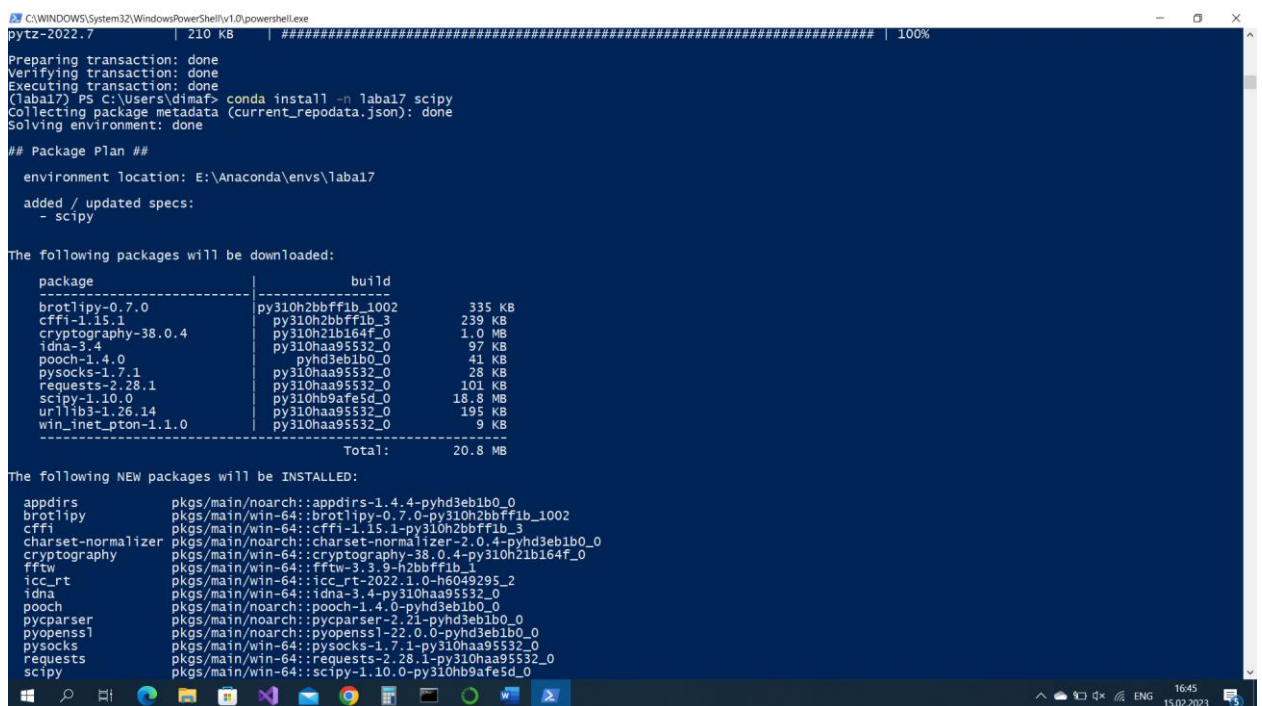


Рисунок 3 – Установка пакетов

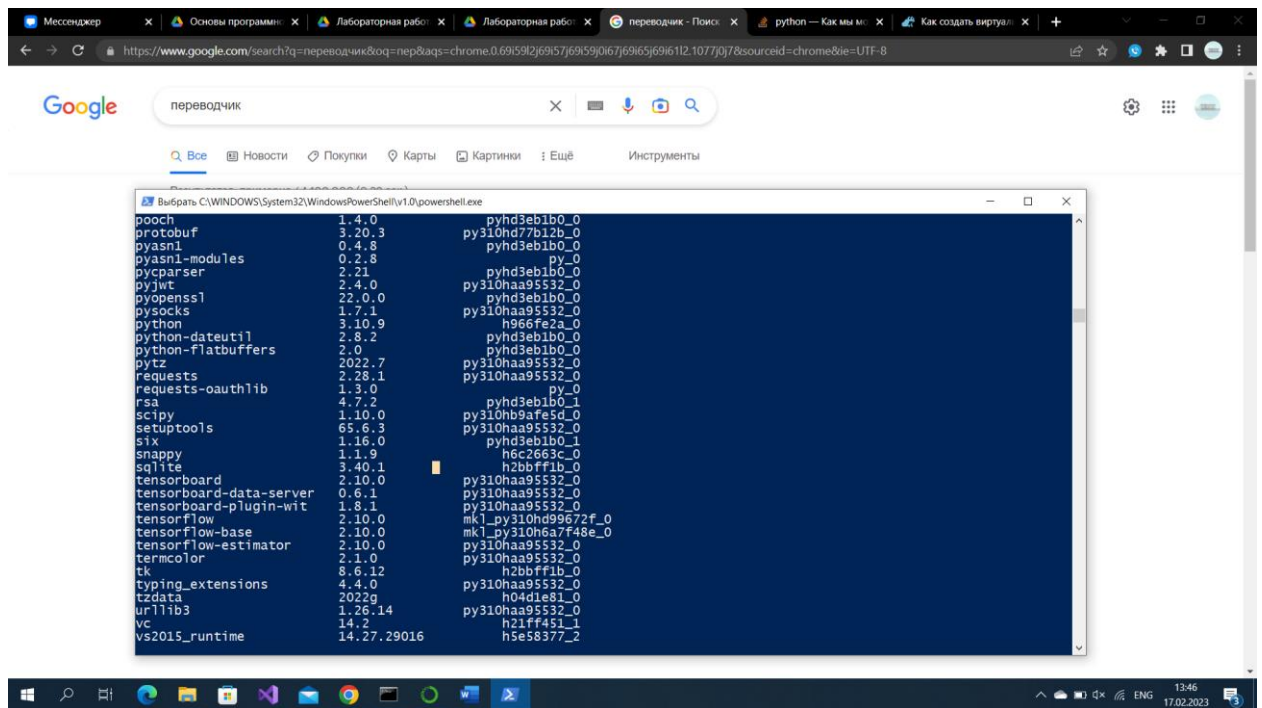


Рисунок 4 – Установленные пакеты

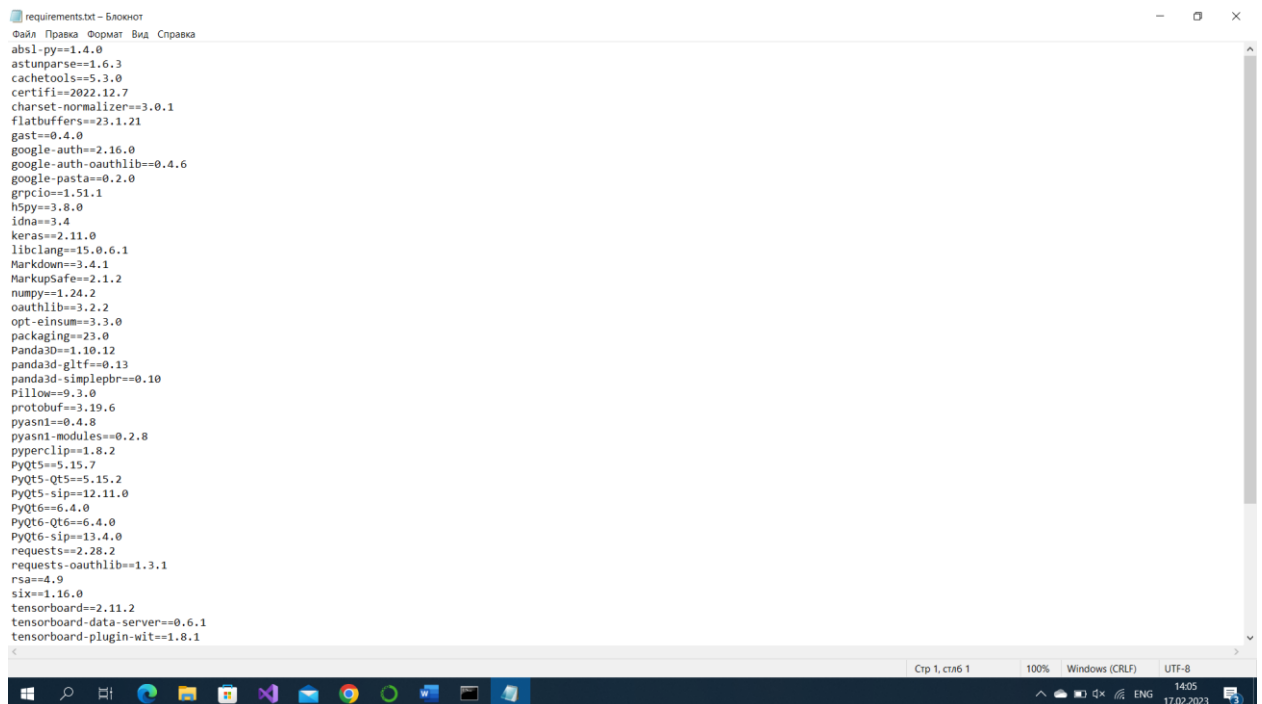


Рисунок 5 – Файл requirements.txt

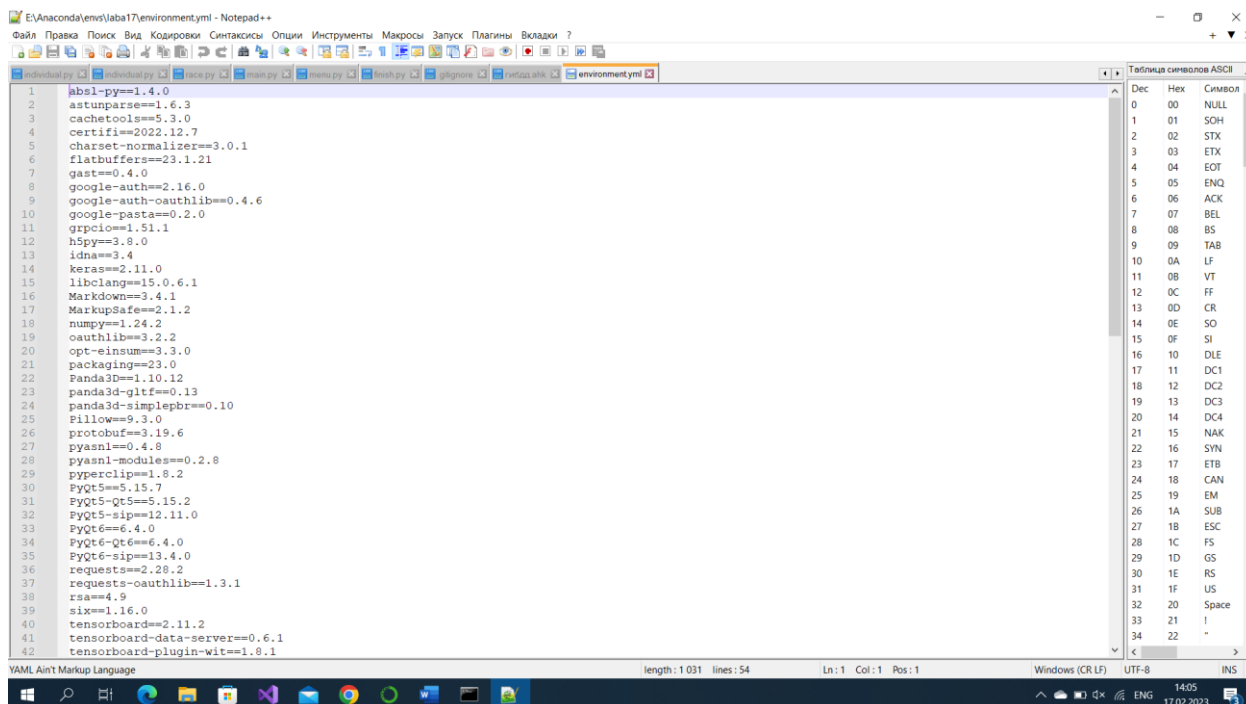


Рисунок 6 – Файл environment.yml

Вывод: приобрели навыки по работе с менеджером пакетов `pip` и виртуальными окружениями с помощью языка программирования Python версии 3.x.

Ответы на контрольные вопросы:

1. Каким способом можно установить пакет Python, не входящий в стандартную библиотеку? Существует так называемый Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач. Там также есть возможность выкладывать свои пакеты. Для скачивания и установки используется специальная утилита, которая называется `pip`.

2. Как осуществить установку менеджера пакетов `pip`? Сначала необходимо скачать скрипт `get-pip.py`:

```
$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

и выполнить его командой а затем выполнить его:

```
$ python get-pip.py
```

3. Откуда менеджер пакетов `pip` по умолчанию устанавливает пакеты? По умолчанию он их скачивает из Python Package Index (PyPI).

4. Как установить последнюю версию пакета с помощью `pip`? Для этого используется команда `$ pip install ProjectName`

5. Как установить заданную версию пакета с помощью `pip`? Для этого используется команда `$ pip install ProjectName==3.2`

6. Как установить пакет из `git` репозитория (в том числе `GitHub`) с помощью `pip`? Для этого используется команда `$ pip install -e git+https://gitrepo.com/ProjectName.git`

7. Как установить пакет из локальной директории с помощью `pip`? Для этого используется команда `$ pip install ./dist/ProjectName.tar.gz`

8. Как удалить установленный пакет с помощью `pip`? Для удаления установленного пакета необходимо воспользоваться командой `$ pip uninstall ProjectName`

9. Как обновить установленный пакет с помощью `pip`? Для обновления установленного пакета необходимо воспользоваться командой `$ pip install --upgrade ProjectName`

10. Как отобразить список установленных пакетов с помощью `pip`? Для этого существует команда `pip list`

11. Каковы причины появления виртуальных окружений в языке `Python`? В системе для интерпретатора `Python` может быть установлена глобально только одна версия пакета. Это порождает ряд проблем:

1. Проблема обратной совместимости
2. Проблема коллективной разработки

Поэтому для каждого проекта нужна своя "песочница", которая изолирует зависимости. Такая "песочница" придумана и называется "виртуальным окружением" или "виртуальной средой".

12. Каковы основные этапы работы с виртуальными окружениями?

Основные этапы работы с виртуальным окружением:

1. Создание
2. Активация
3. Работа с виртуальным окружением
4. Деактивация после окончания работы
5. Удаляем папку