

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №17**  
**дисциплины «Основы программной инженерии»**

Выполнил:  
Джараян Арег Александрович  
2 курс, группа ПИЖ-б-о-22-1,  
09.03.04 «Программная инженерия»,  
направленность (профиль) «Разработка и  
сопровождение программного  
обеспечения», очная форма обучения

\_\_\_\_\_  
(подпись)

Проверил Воронкин Роман Александрович

\_\_\_\_\_  
(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2024 г.

**Тема:** 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.

### 1. Создание нового репозитория с лицензией MIT.

Required fields are marked with an asterisk (\*).


Owner \* / Repository name \*


 aregdz / 17

✔ 17 is available.

Great repository names are short and memorable. Need inspiration? How about **ubiquitous-octo-parakeet** ?

Description (optional)

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: Python

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: MIT License

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1 – Создание репозитория

### 2. Клонировал репозиторий на рабочий ПК.

```
aregd@DESKTOP-5KV9QA9 MINGW64 ~  
$ cd "D:\Рабочий стол\4 семестр\опи\17"  
  
aregd@DESKTOP-5KV9QA9 MINGW64 /d/Рабочий стол/4 семестр/опи/17  
$ git clone "https://github.com/aregdz/17.git"  
Cloning into '17'...  
remote: Enumerating objects: 4, done.  
remote: Counting objects: 100% (4/4), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0  
Receiving objects: 100% (4/4), done.  
  
aregd@DESKTOP-5KV9QA9 MINGW64 /d/Рабочий стол/4 семестр/опи/17  
$ |
```

Рисунок 2 – Клонирование репозитория

3. Дополнил файл .gitignore необходимыми инструкциями.

```
1  # Byte-compiled / optimized / DLL files
2  __pycache__/
3  *.py[cod]
4  *$py.class
5
6  # C extensions
7  *.so
8
9  # Distribution / packaging
10 .Python
11 build/
12 develop-eggs/
13 dist/
14 downloads/
15 eggs/
16 .eggs/
17 lib/
18 lib64/
19 parts/
20 sdist/
21 var/
22 wheels/
23
```

Рисунок 3 – Файл .gitignore

```
aregd@DESKTOP-5KV9QA9 MINGW64 /d/Рабочий стол/4 семестр/опи/17/17 (main)
$ git checkout -b develop
Switched to a new branch 'develop'

aregd@DESKTOP-5KV9QA9 MINGW64 /d/Рабочий стол/4 семестр/опи/17/17 (develop)
$
```

Рисунок 4 – организация ветки

4. Запуск Anaconda Powershell Prompt.

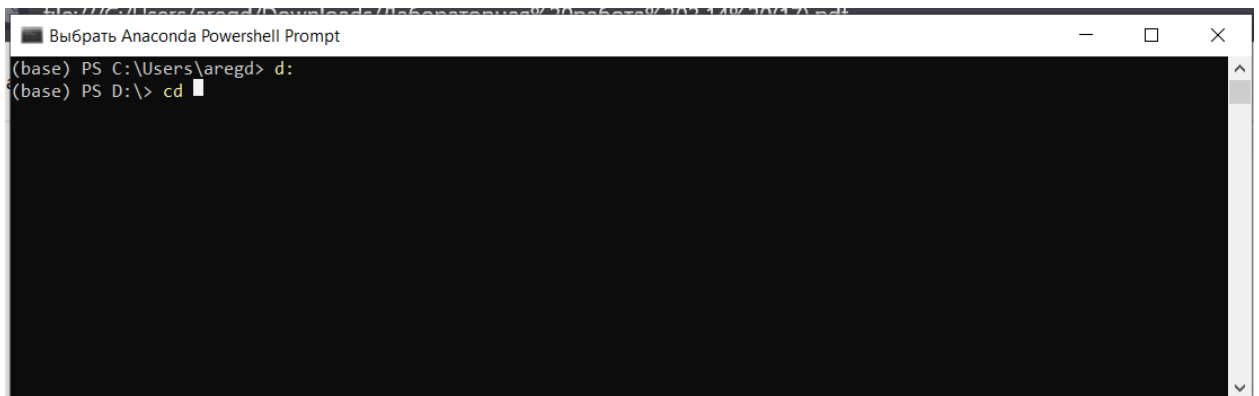


Рисунок 5 – Запуск Anaconda Powershell Prompt

5. Создание виртуального окружения Anaconda с именем репозитория.

```
(base) D:\Рабочий стол\4 семестр\опи\17>conda create -n lab17 python=3.10
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 24.1.1

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=24.1.1
```

Рисунок 6 – Командная строка

6. Активация виртуального окружения, установка пакетов: pip, NumPy, Pandas, SciPy.

```
(base) D:\Рабочий стол\4 семестр\опи\17>conda activate lab17
(lab17) D:\Рабочий стол\4 семестр\опи\17>
```

Рисунок 7 – Активация

```
environment location: C:\Users\aregd\.conda\envs\lab17

added / updated specs:
- numpy
- pandas
- pip
- scipy

The following packages will be downloaded:
```

package	build	
bottleneck-1.3.7	py310h9128911_0	115 KB
intel-openmp-2023.1.0	h59b6b97_46320	2.7 MB
mkl-2023.1.0	h6b88ed4_46358	155.9 MB
mkl-service-2.4.0	py310h2bbff1b_1	44 KB
mkl_fft-1.3.8	py310h2bbff1b_0	170 KB
mkl_random-1.2.4	py310h59b6b97_0	227 KB

Рисунок 8 – Установка пакетов

## 7. Установка tensorflow при помощи conda и при помощи pip.

```
(lab17) D:\Рабочий стол\4 семестр\опи\17>conda install tensorflow
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: unsuccessful attempt using repodata from current_repodata.json, retrying with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 24.1.1

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=24.1.1

## Package Plan ##
```

Рисунок 9 – Установка при помощи conda

```
(lab17) D:\Рабочий стол\4 семестр\опи\17>conda list tensorflow
# packages in environment at C:\Users\aregd\.conda\envs\lab17:
#
# Name                      Version           Build Channel
tensorflow                  2.10.0            mkl_py310hd99672f_0
tensorflow-base             2.10.0            mkl_py310h6a7f48e_0
tensorflow-estimator        2.10.0            py310haa95532_0

(lab17) D:\Рабочий стол\4 семестр\опи\17>
```

Рисунок 11 – Установка при помощи conda

```
(lab17) D:\Рабочий стол\4 семестр\опи\17>pip install tensorflow
Requirement already satisfied: tensorflow in c:\users\aregd\.conda\envs\lab17\lib\site-packages (2.10.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (2.0)
Requirement already satisfied: gast<0.4.0,>=0.2.1 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: h5py>=2.9.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (3.9.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (1.1.2)
Collecting libclang>=13.0.0 (from tensorflow)
  Downloading libclang-16.0.6-py2.py3-none-win_amd64.whl.metadata (5.3 kB)
Requirement already satisfied: numpy>=1.20 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (1.26.3)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (23.1)
Collecting protobuf<3.20,>=3.9.2 (from tensorflow)
  Downloading protobuf-3.19.6-cp310-cp310-win_amd64.whl (895 kB)
----- 895.7/895.7 kB 2.3 MB/s eta 0:00:00
Requirement already satisfied: setuptools in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (68.2.2)
Requirement already satisfied: six>=1.12.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (2.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (4.9.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\aregd\.conda\envs\lab17\lib\site-packages (from tensorflow) (1.14.1)
```

Рисунок 12 – Установка при помощи pip

## 8. Формирование файлов requirement.txt и environment.yml.

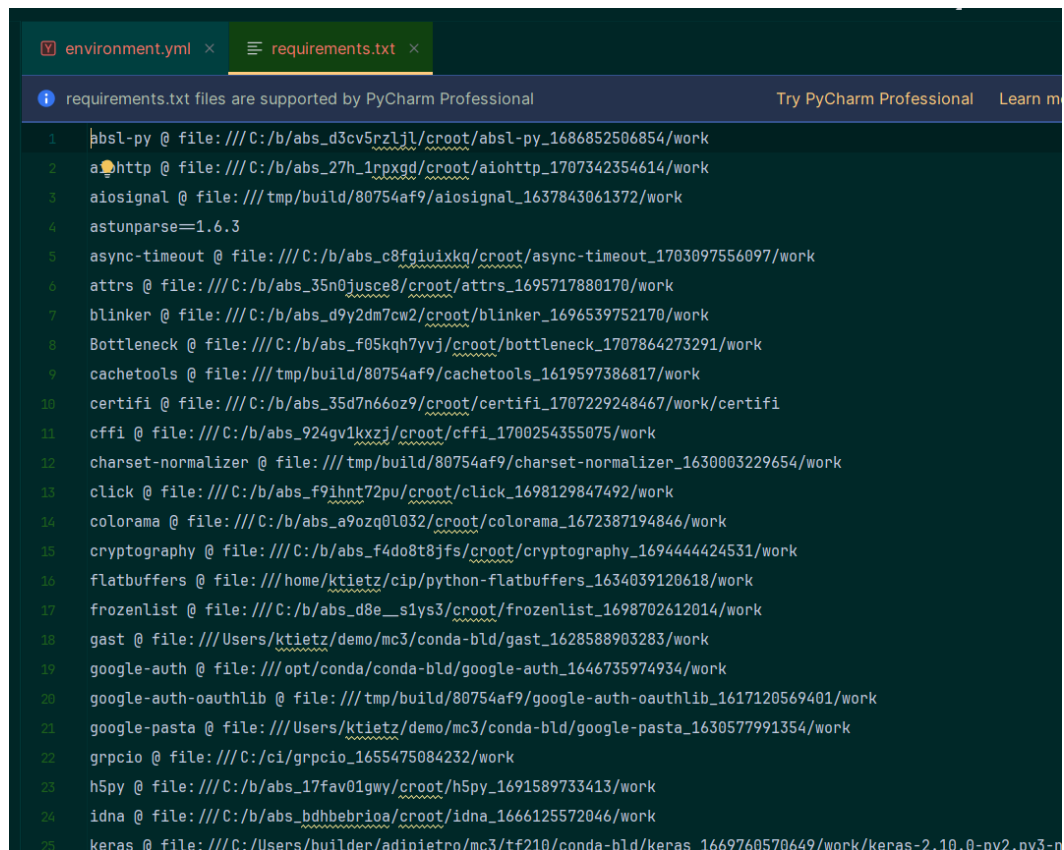
```
(lab17) D:\Рабочий стол\4 семестр\опи\17>pip freeze > requirements.txt
(lab17) D:\Рабочий стол\4 семестр\опи\17>conda env export > environment.yml
```

Рисунок 13 – формирование файлов

```
environment.yml x
1  name: lab17
2  channels:
3    - defaults
4  dependencies:
5    - _tfselect=2.3.0=mkl
6    - absl-py=1.4.0=py310h9a95532_0
7    - aiohttp=3.9.3=py310h2bbff1b_0
8    - aiosignal=1.2.0=pyhd3eb1b0_0
9    - astunparse=1.6.3=py_0
10   - async-timeout=4.0.3=py310h9a95532_0
11   - attrs=23.1.0=py310h9a95532_0
12   - blas=1.0=mkl
13   - blinker=1.6.2=py310h9a95532_0
14   - bottleneck=1.3.7=py310h9128911_0
15   - bzip2=1.0.8=he774522_0
16   - ca-certificates=2023.12.12=h9a95532_0
17   - cachetools=4.2.2=pyhd3eb1b0_0
18   - certifi=2024.2.2=py310h9a95532_0
19   - cffi=1.16.0=py310h2bbff1b_0
20   - charset-normalizer=2.0.4=pyhd3eb1b0_0
21   - click=8.1.7=py310h9a95532_0
22   - colorama=0.4.6=py310h9a95532_0
23   - cryptography=41.0.3=py310h3438e0d_0
24   - flatbuffers=2.0.0=h6c2663c_0
25   - frozenlist=1.4.0=py310h2bbff1b_0
26   - gast=0.4.0=pyhd3eb1b0_0
27   - giflib=5.2.1=h8cc25b3_3
28   - google-auth=2.6.0=pyhd3eb1b0_0
```

Рисунок 14 – environment.yml





```
environment.yml x requirements.txt x
requirements.txt files are supported by PyCharm Professional Try PyCharm Professional Learn more
1  absl-py @ file:///C:/b/abs_d3cv5r3ljl/croot/absl-py_1686852506854/work
2  aiohttp @ file:///C:/b/abs_27h_1rpxgd/croot/aiohttp_1707342354614/work
3  aiosignal @ file:///tmp/build/80754af9/aiosignal_1637843061372/work
4  astunparse==1.6.3
5  async-timeout @ file:///C:/b/abs_c8fgiuvixkq/croot/async-timeout_1703097556097/work
6  attrs @ file:///C:/b/abs_35n0jusce8/croot/attrs_1695717880170/work
7  blinker @ file:///C:/b/abs_d9y2dm7cw2/croot/blinker_1696539752170/work
8  Bottleneck @ file:///C:/b/abs_f05kqh7yvjj/croot/bottleneck_1707864273291/work
9  cachetools @ file:///tmp/build/80754af9/cachetools_1619597386817/work
10 certifi @ file:///C:/b/abs_35d7n66oz9/croot/certifi_1707229248467/work/certifi
11 cffi @ file:///C:/b/abs_924gv1kxzj/croot/cffi_1700254355075/work
12 charset-normalizer @ file:///tmp/build/80754af9/charset-normalizer_1630003229654/work
13 click @ file:///C:/b/abs_f9ihnt72pu/croot/click_1698129847492/work
14 colorama @ file:///C:/b/abs_a9ozq0l032/croot/colorama_1672387194846/work
15 cryptography @ file:///C:/b/abs_f4do8t8jfs/croot/cryptography_1694444424531/work
16 flatbuffers @ file:///home/ktietz/cip/python-flatbuffers_1634039120618/work
17 frozenlist @ file:///C:/b/abs_d8e_s1ys3/croot/frozenlist_1698702612014/work
18 gast @ file:///Users/ktietz/demo/mc3/conda-bld/gast_1628588903283/work
19 google-auth @ file:///opt/conda/conda-bld/google-auth_1646735974934/work
20 google-auth-oauthlib @ file:///tmp/build/80754af9/google-auth-oauthlib_1617120569401/work
21 google-pasta @ file:///Users/ktietz/demo/mc3/conda-bld/google-pasta_1630577991354/work
22 grpcio @ file:///C:/ci/grpcio_1655475084232/work
23 h5py @ file:///C:/b/abs_17fav01gwy/croot/h5py_1691589733413/work
24 idna @ file:///C:/b/abs_bdhhbebr1oa/croot/idna_1666125572046/work
25 keras @ file:///C:/Users/builder/adipietro/mc3/tf210/conda-bld/keras_1669760570649/work/keras-2.10.0-py2.py3-n
```

Рисунок 15 – requirements.txt

#### Контрольные вопросы:

- 1.Если вы хотите добавить новый пакет Python, который не входит в стандартную библиотеку, то просто используйте менеджер пакетов `pip`. Просто запустите команду `pip install имя_пакета`, и пакет будет установлен.
- 2.Для того чтобы получить менеджер пакетов `pip`, просто загрузите скрипт `get-pip.py` и выполните его с помощью Python. Просто запустите `python get-pip.py`.
- 3.По умолчанию менеджер пакетов `pip` загружает пакеты из Python Package Index (PyPI), что делает процесс установки пакетов очень простым.
- 4.Если вам нужна последняя версия пакета, то можно воспользоваться командой `pip install имя_пакета`.
- 5.Для установки конкретной версии пакета, используйте команду `pip install имя_пакета==версия`.
- 6.Установить пакет из `git` репозитория можно с помощью команды `pip install git+https://github.com/пользователь/репозиторий.git`.

7.Если вам нужно установить пакет из локальной директории, просто выполните `pip install ./директория`.

8.Чтобы удалить установленный пакет, просто выполните `pip uninstall имя_пакета`.

9.Для обновления установленного пакета, используйте команду `pip install --upgrade имя_пакета`.

10.Список всех установленных пакетов можно увидеть с помощью команды `pip list`.

11.Виртуальные окружения в Python позволяют изолировать зависимости проектов и предотвращать конфликты между разными версиями пакетов.

12.Создание, активация, установка зависимостей и деактивация виртуальных окружений - основные этапы работы с ними.

13.Для работы с виртуальными окружениями через `venv`, используйте команды: создание - `python -m venv имя_окружения`, активация - `source имя_окружения/bin/activate` или `имя_окружения\Scripts\activate`, установка и деактивация.

14.Работа с виртуальными окружениями через `virtualenv` аналогична, но требует предварительной установки `virtualenv`.

15.Виртуальные окружения `pipenv` включают создание, активацию, установку зависимостей через `pipenv install`, активацию через `pipenv shell` и установку зависимостей из `Pipfile`.

16.Файл `requirements.txt` используется для определения зависимостей проекта с указанием версий пакетов.

17.Преимущества `conda` включают управление не только Python-пакетами, но и бинарными зависимостями и окружениями.

18.Пакетный менеджер `conda` поставляется в дистрибутивах Python Anaconda и Miniconda.

19.Виртуальное окружение `conda` создается командой `conda create --name имя_окружения пакеты`.

20.Для активации виртуального окружения conda используйте conda activate имя\_окружения, для установки пакетов - conda install пакеты.

21.Для деактивации виртуального окружения conda используйте conda deactivate, для удаления - conda remove --name имя\_окружения --all.

22.Файл environment.yml используется для определения окружения conda и его зависимостей.

23.Создать виртуальное окружение conda из файла environment.yml можно с помощью conda env create -f environment.yml.

24.В PyCharm работа с виртуальными окружениями conda включает выбор окружения при настройке проекта и управление зависимостями.

25.Файлы requirements.txt и environment.yml необходимы для воспроизводимости среды разработки и зависимостей проекта среди разработчиков.