

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 №2.14**  
**дисциплины «Основы программной инженерии»**

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

---

(подпись)

Преподаватель:  
Воронкин Р.А., канд. тех. наук, доцент,  
доцент кафедры инфокоммуникаций

---

(подпись)

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

Ставрополь, 2024 г.

**Тема:** Установка пакетов в Python. Виртуальные окружения.

**Цель:** приобретение навыков по работе с модулями и пакетами языка программирования Python версии 3.x.

### Результаты выполнения

1. Создали репозиторий с лицензией MIT, добавили в .gitignore необходимые правила для работы с IDE PyCharm, клонировали репозиторий, организовали репозиторий в соответствии с моделью git-flow.

```
Last login: Tue Oct 24 20:29:02 on ttys000
itssyoma@MacBook-Air-Sema Основы программной инженерии % git clone https://github.com/itssyoma/megarepo_21.git
Cloning into 'megarepo_21'...
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.
itssyoma@MacBook-Air-Sema Основы программной инженерии % cd megarepo_21
itssyoma@MacBook-Air-Sema megarepo_21 % git checkout -b develop
Switched to a new branch 'develop'
itssyoma@MacBook-Air-Sema megarepo_21 % git branch release
itssyoma@MacBook-Air-Sema megarepo_21 % git branch develop
fatal: a branch named 'develop' already exists
itssyoma@MacBook-Air-Sema megarepo_21 % git branch hotfix
itssyoma@MacBook-Air-Sema megarepo_21 % git branch feature
itssyoma@MacBook-Air-Sema megarepo_21 %
```

Рисунок 1 – Работа с репозиторием в командной строке.

2. Создали виртуальное окружение Anaconda с именем репозитория.

```
(base) PS Z:\Yandex.Disk.localized\Учеба\4 семестр\Основы программной инженерии\megarepo_214> conda create -n megarepo_214 python=3.11.7
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\itssyoma\.conda\envs\megarepo_214

added / updated specs:
  - python=3.11.7

The following packages will be downloaded:

package | build | size
-----|-----|-----
bzip2-1.0.8 | h2bbff1b_5 | 78 KB
ca-certificates-2024.3.11 | haa95532_0 | 128 KB
tzdata-2024a | h04d1e81_0 | 116 KB
xz-5.4.6 | h8cc25b3_0 | 587 KB
-----|-----|-----
Total: | | 910 KB
```

Рисунок 2.1 – Создание чистого conda-окружения

```
и\megarepo_214> conda activate megarepo_214
инженерии\megarepo_214> _
```

Рисунок 2.2 – Активация conda-окружения

3. Установили в виртуальное окружение следующие пакеты: pip, NumPy, Pandas, SciPy.

```
(megarepo_214) PS Z:\Yandex.Disk.localized\Учеба\4 семестр\Основы программной инженерии\megarepo_214> conda install pip, NumPy, Pandas, SciPy
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\itssyoma\.conda\envs\megarepo_214

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

Рисунок 3 – Установка пакетов

#### 4. Установили пакет TensorFlow.

```
Solving environment: / warning libmamba Added empty dependency for problem type SOLVER_RULE_UPDATE
failed

LibMambaUnsatisfiableError: Encountered problems while solving:
- nothing provides bleach 1.5.0 needed by tensorboard-1.7.0-py35he025d50_1

Could not solve for environment specs
The following packages are incompatible
├─ pin-1 is installable and it requires
│   └─ python 3.11.* , which can be installed;
└─ tensorflow is not installable because there are no viable options
    ├── tensorflow [1.10.0|1.9.0] would require
    │   ├── python 3.5.* , which conflicts with any installable versions previously reported;
    │   ├── tensorflow [1.10.0|1.11.0|...|2.1.0] would require
    │   │   ├── python 3.6.* , which conflicts with any installable versions previously reported;
    │   │   ├── tensorflow [1.13.1|1.14.0|...|2.9.1] would require
    │   │   │   ├── python 3.7.* , which conflicts with any installable versions previously reported;
    │   │   │   ├── tensorflow [1.7.0|1.7.1|1.8.0] would require
    │   │   │   │   ├── tensorboard [ >=1.7.0,<1.8.0 | >=1.8.0,<1.9.0 ], which requires
    │   │   │   │   │   ├── bleach 1.5.0 , which does not exist (perhaps a missing channel);
    │   │   │   │   ├── tensorflow [2.10.0|2.8.2|2.9.1] would require
    │   │   │   │   │   ├── python 3.10.* , which conflicts with any installable versions previously reported;
    │   │   │   │   ├── tensorflow [2.10.0|2.3.0|...|2.9.1] would require
    │   │   │   │   │   ├── python 3.8.* , which conflicts with any installable versions previously reported;
    │   │   │   │   ├── tensorflow [2.10.0|2.5.0|2.6.0|2.8.2|2.9.1] would require
    │   │   │   │   │   ├── python 3.9.* , which conflicts with any installable versions previously reported.
```

Рисунок 4.1 – Попытка установки TensorFlow менеджером пакетов conda.

При попытке установить TensorFlow через менеджер пакетов conda получили ошибку о том, что для текущей версии Python невозможно установить пакет. Это связано с недоступностью TensorFlow в официальных репозиториях conda: TensorFlow может не быть включен в стандартные пакеты, предлагаемые conda, из-за чего его нельзя установить напрямую через conda.

Попробуем установить TensorFlow напрямую из Python Package Index (PyPI) через утилиту pip.

```
(megarepo_214) PS Z:\Yandex.Disk.localized\Учеба\4 семестр\Основы программной инженерии\megarepo_214> pip install TensorFlow
```

Рисунок 4.2 – Попытка установки TensorFlow менеджером пакетов pip.

```
Successfully installed MarkupSafe-2.1.5 TensorFlow-2.16.1 absl-py-2.1.0 astunparse-1.6.3 certifi-2024.2.2 charset-normalizer-3.3.2 flatbuffers-24.3.7 gast-0.5.4 google-pasta-0.2.0 grpcio-1.62.1 h5py-3.10.0 idna-3.6 keras-3.1.0 libclang-18.1.1 markdown-3.6 markdown-it-py-3.0.0 mdurl-0.1.2 ml-dtypes-0.3.2 namex-0.0.7 opt-einsum-3.3.0 optree-0.10.0 packaging-24.0 protobuf-4.25.3 pygments-2.17.2 requests-2.31.0 rich-13.7.1 tensorboard-2.16.2 tensorboard-data-server-0.7.2 tensorflow-intel-2.16.1 tensorflow-io-gcs-filesystem-0.31.0 termcolor-2.4.0 typing-extensions-4.10.0 urllib3-2.2.1 werkzeug-3.0.1 wrapt-1.16.0
```

Рисунок 4.3 – Успешная попытка установки.

## 5. Сформировали файлы requirements.txt и environment.yml.

```
pip freeze > requirements.txt
conda env export > environment.yml
```

Рисунок 5.1 – формирование файлов requirements.txt и environment.yml

Сравним содержание обоих файлов.

```
1 name: megarepo_214
2 channels:
3   - defaults
4 dependencies:
5   - blas=1.0=mkl
6   - bottleneck=1.3.7=py311hd7041d2_0
7   - bzip2=1.0.8=h2bbff1b_5
8   - ca-certificates=2024.3.11=haa95532_0
9   - icc_rt=2022.1.0=h6049295_2
10  - intel-openmp=2023.1.0=h59b6b97_46320
11  - libffi=3.4.4=hd77b12b_0
12  - mkl=2023.1.0=h6b88ed4_46358
13  - mkl-service=2.4.0=py311h2bbff1b_1
14  - mkl_fft=1.3.8=py311h2bbff1b_0
15  - mkl_random=1.2.4=py311h59b6b97_0
16  - numexpr=2.8.7=py311h1fcade_0
17  - numpy=1.26.4=py311hdab7c0b_0
18  - numpy-base=1.26.4=py311hd01c5d8_0
19  - openssl=3.0.13=h2bbff1b_0
20  - pandas=2.2.1=py311hea22821_0
21  - pip=23.3.1=py311haa95532_0
22  - python=3.11.7=he1021f5_0
23  - python-dateutil=2.8.2=pyhd3eb1b0_0
24  - python-tzdata=2023.3=pyhd3eb1b0_0
25  - pytz=2023.3.post1=py311haa95532_0
26  - scipy=1.11.4=py311hc1ccb85_0
27  - setuptools=68.2.2=py311haa95532_0
28  - six=1.16.0=pyhd3eb1b0_1
29  - sqlite=3.41.2=h2bbff1b_0
30  - tbb=2021.8.0=h59b6b97_0
31  - tk=8.6.12=h2bbff1b_0
32  - tzdata=2024a=h04d1e81_0
33  - vc=14.2=h21ffa51_1
34  - vs2015_runtime=14.2.29016=h5e58377_2
35  - wheel=0.41.2=py311haa95532_0
36  - xz=5.4.6=h8cc25b3_0
37  - zlib=1.2.13=h8cc25b3_0
38  - pip:
39    - absl-py==2.1.0
40    - astunparse==1.6.3
41    - certifi==2024.2.2
42    - charset-normalizer==3.3.2
43    - flatbuffers==24.3.7
44    - gast==0.5.4
45    - google-pasta==0.2.0
46    - grpcio==1.62.1
47    - h5py==3.10.0
48    - idna==3.6
49    - keras==3.1.0
50    - libclang==18.1.1
51    - markdown-it-py==3.0.0
52    - markdown-it-py==3.0.0
53    - markupsafe==2.1.5
54    - mdurl==0.1.2
55    - ml-dtypes==0.3.2
56    - nameex==0.0.7
57    - opt-einsum==3.3.0
58    - optree==0.10.0
59    - packaging==24.0
60    - protobuf==4.25.3
61    - pygments==2.17.2
62    - requests==2.31.0
63    - rich==13.7.1
64    - tensorboard==2.16.2
65    - tensorboard-data-server==0.7.2
66    - tensorflow==2.16.1
67    - tensorflow-intel==2.16.1
68    - tensorflow-io-gcs-filesystem==0.31.0
69    - termcolor==2.4.0
70    - typing-extensions==4.10.0
71    - urllib3==2.2.1
72    - werkzeug==3.0.1
73    - wrapt==1.16.0
74 prefix: C:\Users\itssyoma\.conda\envs\megarepo_214

1 absl-py==2.1.0
2 astunparse==1.6.3
3 Bottleneck @ file:///C:/b/abs_f05kqh7yvj/croot/bottleneck_1707864273291/work
4 certifi==2024.2.2
5 charset-normalizer==3.3.2
6 flatbuffers==24.3.7
7 gast==0.5.4
8 google-pasta==0.2.0
9 grpcio==1.62.1
10 h5py==3.10.0
11 idna==3.6
12 keras==3.1.0
13 libclang==18.1.1
14 Markdown==3.6
15 markdown-it-py==3.0.0
16 MarkupSafe==2.1.5
17 mdurl==0.1.2
18 mkl-fft @ file:///C:/b/abs_19i1y8ykas/croot/mkl_fft_1695058226480/work
19 mkl-random @ file:///C:/b/abs_edwkj1_o69/croot/mkl_random_1695059866750/work
20 mkl-service==2.4.0
21 ml-dtypes==0.3.2
22 nameex==0.0.7
23 numexpr @ file:///C:/b/abs_5fucrt5dc/croot/numexpr_1696515448831/work
24 numpy @
   file:///C:/b/abs_clywpu18az/croot/numpy_and_numpy_base_1708638681471/work/dist/numpy-1.26
   .4-cp311-cp311-win_amd64
   whl#sha256=5df03e04dc1c2826d3f404fcd7f93c097901f5da9b91f4f394f79d4e038ed81d
25 opt-einsum==3.3.0
26 optree==0.10.0
27 packaging==24.0
28 pandas @
   file:///C:/b/abs_3awk0iw2ab/croot/pandas_1709590545218/work/dist/pandas-2.2
   .1-cp311-cp311-win_amd64
   whl#sha256=d6c6b110e8e5c8a299efcaa9935f84d8ad7abc5511d6d8ab94cc6bd137d93ff3
29 protobuf==4.25.3
30 Pygments==2.17.2
31 python-dateutil @ file:///tmp/build/80754af9/python-dateutil_1626374649649/work
32 pytz @ file:///C:/b/abs_19q3ljkeza/croot/pytz_1695131651401/work
33 requests==2.31.0
34 rich==13.7.1
35 scipy==1.11.4
36 six @ file:///tmp/build/80754af9/six_1644875935823/work
37 tensorboard==2.16.2
38 tensorboard-data-server==0.7.2
39 tensorflow==2.16.1
40 tensorflow-intel==2.16.1
41 tensorflow-io-gcs-filesystem==0.31.0
42 termcolor==2.4.0
43 typing_extensions==4.10.0
44 tzdata @ file:///croot/python-tzdata_1690578112552/work
45 urllib3==2.2.1
46 Werkzeug==3.0.1
47 wrapt==1.16.0
```

Рисунок 5.2 – Содержание файла environment.yml

Рисунок 5.3 – Содержание файла requirements.txt

Список зависимостей, сохраняемый conda, содержит в себе больше пакетов, чем pip, и включает в себя пакеты, устанавливаемые по умолчанию при создании виртуального окружения. Пакеты, установленные через pip, выделяются отдельным блоком. Также файл environment.yml содержит в себе название виртуального окружения, репозитории загрузки пакетов и расположение виртуального окружения (папки env). Файл requirements.txt более емкий и не содержит пакетов, устанавливаемых conda автоматически, и так же не содержит самого пакета pip.

### **Ответы на контрольные вопросы**

1. Пакет Python, не входящий в стандартную библиотеку, можно установить с помощью менеджера пакетов pip. Например, для установки пакета requests: `pip install requests`.
2. Для установки менеджера пакетов pip, необходимо скачать и запустить скрипт get-pip.py: `python get-pip.py`.
3. Менеджер пакетов pip по умолчанию устанавливает пакеты из Python Package Index (PyPI).
4. Для установки последней версии пакета с помощью pip: `pip install package_name --upgrade`.
5. Для установки заданной версии пакета с помощью pip: `pip install package_name==version`.
6. Для установки пакета из git репозитория с помощью pip: `pip install git+https://github.com/user/repo.git`.
7. Для установки пакета из локальной директории с помощью pip: `pip install /path/to/package`.
8. Для удаления установленного пакета с помощью pip: `pip uninstall package_name`.
9. Для обновления установленного пакета с помощью pip: `pip install package_name --upgrade`.

10. Для отображения списка установленных пакетов с помощью `pip`: `pip list`.
11. Виртуальные окружения в Python используются для изоляции проектов и их зависимостей друг от друга.
12. Основные этапы работы с виртуальными окружениями: создание, активация, установка пакетов, деактивация, удаление.
13. Работа с виртуальными окружениями с помощью `venv` осуществляется через команды в терминале, например: `python -m venv myenv`.
14. Работа с виртуальными окружениями с помощью `virtualenv` аналогична работе с `venv`, но используется команда `virtualenv myenv`.
15. Работа с виртуальными окружениями `pipenv` осуществляется через команды в терминале, например: `pipenv install package_name`.
16. Файл `requirements.txt` используется для хранения списка зависимостей проекта. Создать его можно командой `pip freeze > requirements.txt`, формат - каждая строка содержит название и версию пакета.
17. Преимущества пакетного менеджера `conda` по сравнению с `pip`: управление зависимостями, виртуальные окружения и установка бинарных пакетов.
18. Пакетный менеджер `conda` входит в дистрибутивы `Anaconda` и `Miniconda`.
19. Для создания виртуального окружения `conda`: `conda create --name myenv`.
20. Для активации и установки пакетов в виртуальное окружение `conda`: `conda activate myenv` и затем `conda install package_name`.
21. Для деактивации и удаления виртуального окружения `conda`: `conda deactivate` и затем `conda remove --name myenv --all`.
22. Файл `environment.yml` используется для описания окружения `conda`, создать его можно в текстовом редакторе.

23. Для создания виртуального окружения conda с помощью файла environment.yml: `conda env create -f environment.yml`.

24. В PyCharm можно работать с виртуальными окружениями conda через настройки проекта (Project Interpreter).

25. Файлы requirements.txt и environment.yml должны храниться в репозитории git для обеспечения однородности окружения разработчиков и легкости воссоздания проекта на других машинах.