

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 практической работе №2.14**  
**Дисциплины «Программирование на Python»**

Выполнил:

Пустяков Андрей Сергеевич

2 курс, группа ИВТ-б-о-22-1,

09.03.01 «Информатика и  
вычислительная техника (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных  
систем», очная форма обучения

---

(подпись)

Руководитель практики:

Воронкин Р. А. кандидат технических  
наук, доцент, доцент кафедры  
инфокоммуникаций

---

(подпись)

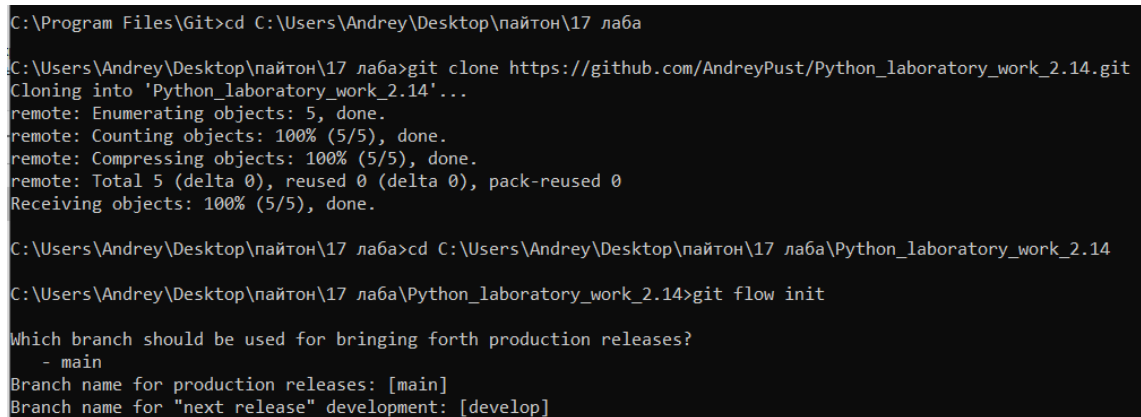
Ставрополь, 2023 г.

Тема: Установка пакетов в Python, виртуальные окружения.

Цель: приобрести навыки по работе с менеджером пакетов «pip» и виртуальными окружениями языка программирования Python версии 3.x.

Ход работы:

Создание общедоступного репозитория на «GitHub», клонирование репозитория, редактирование файла «.gitignore», организация репозитория согласно модели ветвления «git-flow» (рис. 1).



```
C:\Program Files\Git>cd C:\Users\Andrey\Desktop\пайтон\17 лаба
C:\Users\Andrey\Desktop\пайтон\17 лаба>git clone https://github.com/AndreyPust/Python_laboratory_work_2.14.git
Cloning into 'Python_laboratory_work_2.14'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

C:\Users\Andrey\Desktop\пайтон\17 лаба>cd C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14
C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
```

Рисунок 1 – Организация модели ветвления «git-flow».

Создание виртуального окружения в папке с именем репозитория (для примера окружение создавалось в отдельной папке репозитория) (рис. 2).

```
Anaconda Prompt
(base) C:\Users\Andrey>cd C:\Users\Andrey\Desktop\пайтон\17 лаба
(base) C:\Users\Andrey\Desktop\пайтон\17 лаба>cd C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\exam
ple_environment
(base) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>conda create -n Python_lab
oratory_work_2.14
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 23.11.0

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=23.11.0

## Package Plan ##

  environment location: C:\Users\Andrey\anaconda3\envs\Python_laboratory_work_2.14

Proceed ([y]/n)? y
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#   $ conda activate Python_laboratory_work_2.14
#
# To deactivate an active environment, use
#
#   $ conda deactivate
#
(base) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>
```

Рисунок 2 – Создание окружения в папке репозитория.

Установка в созданном виртуальном окружении пакетов «pip» (команда «conda install pip»), «NumPy» («pip install NumPy»), «Pandas» («pip install Pandas»), «SciPy» («pip install SciPy») и проверка списка установленных пакетов (рис. 4, 5, 6, 7, 8).

Установка менеджера пакетов «pip» (рис. 3).

```
Командная строка
Microsoft Windows [Version 10.0.19045.3803]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\Andrey>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2573k  100 2573k    0     0 1322k      0  0:00:01  0:00:01 --:--:-- 1322k

C:\Users\Andrey>python get-pip.py
Collecting pip
  Downloading pip-23.3.2-py3-none-any.whl.metadata (3.5 kB)
  Downloading pip-23.3.2-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 2.9 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.2.1
    Uninstalling pip-23.2.1:
      Successfully uninstalled pip-23.2.1
Successfully installed pip-23.3.2

C:\Users\Andrey>
```

Рисунок 3 – Установка менеджера пакетов «pip».

```
Anaconda Prompt - conda install pip

-----
ca-certificates-2023.12.12 | haa95532_0 | 127 KB
expat-2.5.0 | hd77b12b_0 | 225 KB
openssl-3.0.12 | h2bbff1b_0 | 7.4 MB
pip-23.3.1 | py312haa95532_0 | 2.9 MB
python-3.12.0 | h1d929f7_0 | 16.2 MB
setuptools-68.2.2 | py312haa95532_0 | 1.2 MB
tzdata-2023d | h04d1e81_0 | 117 KB
wheel-0.41.2 | py312haa95532_0 | 150 KB
xz-5.4.5 | h8cc25b3_0 | 593 KB
-----
Total: 28.9 MB

The following NEW packages will be INSTALLED:

bzip2 | pkgs/main/win-64::bzip2-1.0.8-he774522_0
ca-certificates | pkgs/main/win-64::ca-certificates-2023.12.12-haa95532_0
expat | pkgs/main/win-64::expat-2.5.0-hd77b12b_0
libffi | pkgs/main/win-64::libffi-3.4.4-hd77b12b_0
openssl | pkgs/main/win-64::openssl-3.0.12-h2bbff1b_0
pip | pkgs/main/win-64::pip-23.3.1-py312haa95532_0
python | pkgs/main/win-64::python-3.12.0-h1d929f7_0
setuptools | pkgs/main/win-64::setuptools-68.2.2-py312haa95532_0
sqlite | pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_0
tk | pkgs/main/win-64::tk-8.6.12-h2bbff1b_0
tzdata | pkgs/main/noarch::tzdata-2023d-h04d1e81_0
vc | pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime | pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wheel | pkgs/main/win-64::wheel-0.41.2-py312haa95532_0
xz | pkgs/main/win-64::xz-5.4.5-h8cc25b3_0
zlib | pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
list
Package Version
-----
pip 23.3.1
setuptools 68.2.2
wheel 0.41.2
```

Рисунок 4 – Установка пакета «pip» и проверка версии пакета «pip list».

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
install NumPy
Collecting NumPy
  Downloading numpy-1.26.3-cp312-cp312-win_amd64.whl.metadata (61 kB)
----- 61.2/61.2 kB 326.7 kB/s eta 0:00:00
Downloading numpy-1.26.3-cp312-cp312-win_amd64.whl (15.5 MB)
----- 15.5/15.5 MB 2.6 MB/s eta 0:00:00
Installing collected packages: NumPy
Successfully installed NumPy-1.26.3
```

Рисунок 5 – Установка пакета «NumPy».

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
install Pandas
Collecting Pandas
  Downloading pandas-2.1.4-cp312-cp312-win_amd64.whl.metadata (18 kB)
Requirement already satisfied: numpy<2,>=1.26.0 in c:\users\andrey\anaconda3\envs\python_laboratory_work_2.14\lib\site-p
ackages (from Pandas) (1.26.3)
Collecting python-dateutil>=2.8.2 (from Pandas)
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
----- 247.7/247.7 kB 1.4 MB/s eta 0:00:00
Collecting pytz>=2020.1 (from Pandas)
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.1 (from Pandas)
  Downloading tzdata-2023.4-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting six>=1.5 (from python-dateutil>=2.8.2->Pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading pandas-2.1.4-cp312-cp312-win_amd64.whl (10.5 MB)
----- 10.5/10.5 MB 5.7 MB/s eta 0:00:00
Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
----- 502.5/502.5 kB 6.2 MB/s eta 0:00:00
Downloading tzdata-2023.4-py2.py3-none-any.whl (346 kB)
----- 346.6/346.6 kB 5.4 MB/s eta 0:00:00
Installing collected packages: pytz, tzdata, six, python-dateutil, Pandas
Successfully installed Pandas-2.1.4 python-dateutil-2.8.2 pytz-2023.3.post1 six-1.16.0 tzdata-2023.4
```

Рисунок 6 – Установка пакета «Pandas».

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
install SciPy
Collecting SciPy
  Downloading scipy-1.11.4-cp312-cp312-win_amd64.whl.metadata (60 kB)
----- 60.4/60.4 kB 355.8 kB/s eta 0:00:00
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in c:\users\andrey\anaconda3\envs\python_laboratory_work_2.14\lib\s
ite-packages (from SciPy) (1.26.3)
Downloading scipy-1.11.4-cp312-cp312-win_amd64.whl (43.7 MB)
----- 43.7/43.7 MB 5.5 MB/s eta 0:00:00
Installing collected packages: SciPy
Successfully installed SciPy-1.11.4
```

Рисунок 7 – Установка пакета «SciPy».

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
list
Package      Version
-----
numpy        1.26.3
pandas       2.1.4
pip          23.3.1
python-dateutil 2.8.2
pytz         2023.3.post1
scipy        1.11.4
setuptools   68.2.2
six          1.16.0
tzdata       2023.4
wheel        0.41.2
```

Рисунок 8 – Просмотр списка всех установленных пакетов.

Попытка установить пакет «TensorFlow» с помощью менеджера пакетов «conda» оказалась безуспешной (рис. 9). Согласно ошибке, проблема возникает в версиях Python. Самая последняя поддерживаемая версия Python

– это 3.10, а установленная версия – 3.12. Для установки пакета «TensorFlow» была проведена попытка установки с использованием менеджера пакетов «pip» (рис. 10).

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>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: unsuccessful initial attempt using frozen solve. Retrying with flexible solve.
Solving environment: |
Found conflicts! Looking for incompatible packages.
This can take several minutes. Press CTRL-C to abort.
failed

UnsatisfiableError: The following specifications were found
to be incompatible with the existing python installation in your environment:

Specifications:

- tensorflow -> python[version='3.10.*|3.9.*|3.8.*|3.7.*|3.6.*|3.5.*']

Your python: python=3.12

If python is on the left-most side of the chain, that's the version you've asked for.
When python appears to the right, that indicates that the thing on the left is somehow
not available for the python version you are constrained to. Note that conda will not
change your python version to a different minor version unless you explicitly specify
that.
```

Рисунок 9 – Попытка установить «TensorFlow» через менеджер пакетов «conda».

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
install TensorFlow
ERROR: Could not find a version that satisfies the requirement TensorFlow (from versions: none)
ERROR: No matching distribution found for TensorFlow
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>
```

Рисунок 10 – Попытка установить «TensorFlow» через менеджер пакетов «pip».

При установке пакета «TensorFlow» через менеджер пакетов «pip» также возникла ошибка установки пакета. Ошибка скорее всего заключается в использовании не той версии Python. Для решения данной проблемы была переустановлена версия Python на более низкую (рис. 11).

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>conda install 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
```

Рисунок 11 – Смена версии Python.

Установка пакета «TensorFlow» через менеджер пакетов «pip» (рис. 12).

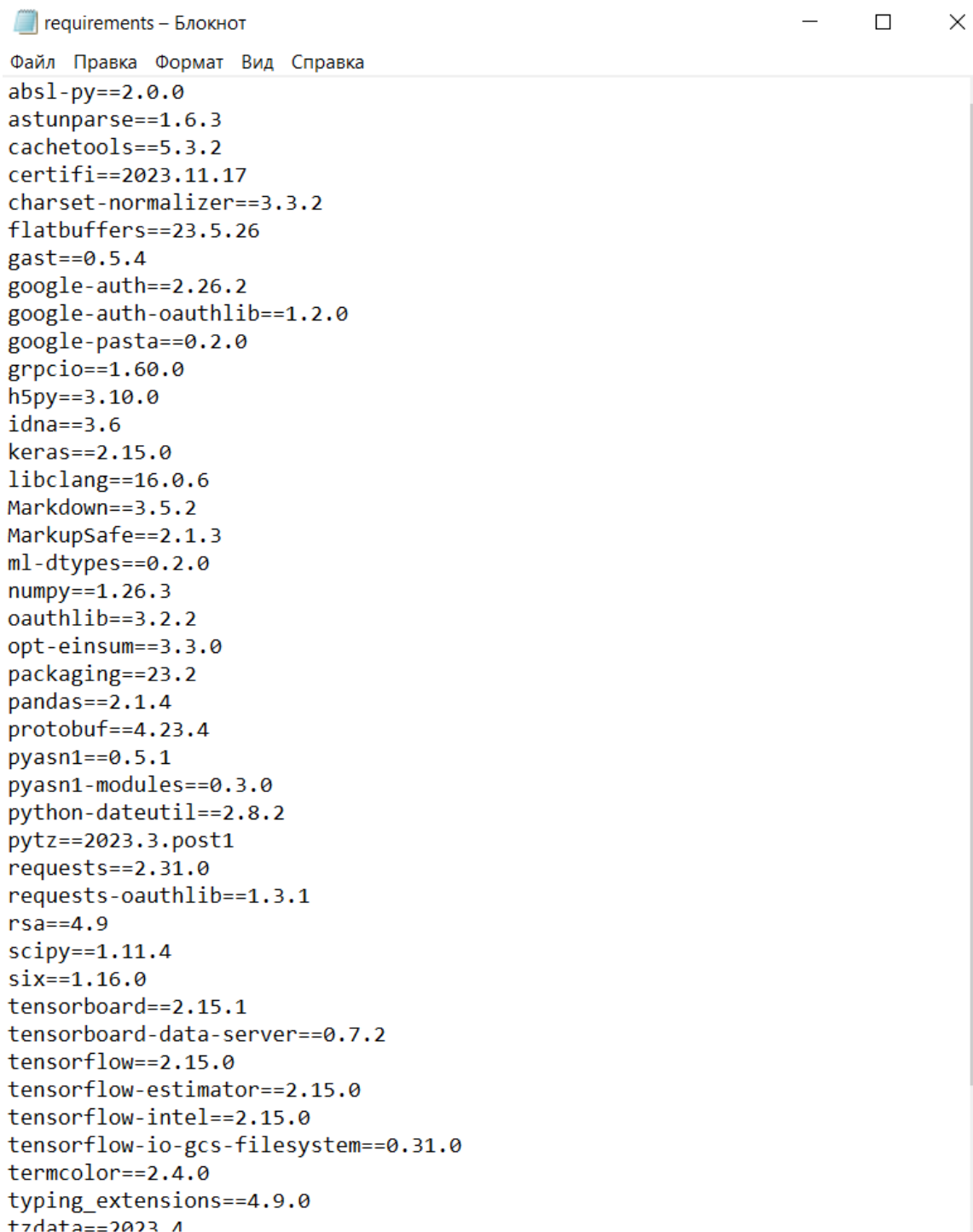
```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
install TensorFlow
Collecting TensorFlow
  Downloading tensorflow-2.15.0-cp310-cp310-win_amd64.whl.metadata (3.6 kB)
Collecting tensorflow-intel==2.15.0 (from TensorFlow)
  Downloading tensorflow_intel-2.15.0-cp310-cp310-win_amd64.whl.metadata (5.1 kB)
Collecting absl-py>=1.0.0 (from tensorflow-intel==2.15.0->TensorFlow)
  Downloading absl_py-2.0.0-py3-none-any.whl.metadata (2.3 kB)
Collecting astunparse>=1.6.0 (from tensorflow-intel==2.15.0->TensorFlow)
  Downloading astunparse-1.6.3-py2.py3-none-any.whl (12 kB)
Collecting flatbuffers>=23.5.26 (from tensorflow-intel==2.15.0->TensorFlow)
  Downloading flatbuffers-23.5.26-py2.py3-none-any.whl.metadata (850 bytes)
Collecting gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 (from tensorflow-intel==2.15.0->TensorFlow)
  Downloading gast-0.5.4-py3-none-any.whl (19 kB)
Collecting google-pasta>=0.1.1 (from tensorflow-intel==2.15.0->TensorFlow)
  Downloading google_pasta-0.2.0-py3-none-any.whl (57 kB)
----- 57.5/57.5 kB 761.8 kB/s eta 0:00:00
```

Рисунок 12 – Установка пакета «TensorFlow».

Формирование файла «requirements.txt» (создание списка зависимостей в текстовом файле) с использованием команды «`pip freeze > requirements.txt`» (перенаправление списка в файл) (рис. 13) и содержимое этого файла (рис. 14).

```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>pip
freeze > requirements.txt
```

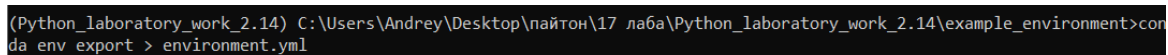
Рисунок 13 – Создание файла «requirements.txt».



```
requirements - Блокнот
Файл  Правка  Формат  Вид  Справка
absl-py==2.0.0
astunparse==1.6.3
cachetools==5.3.2
certifi==2023.11.17
charset-normalizer==3.3.2
flatbuffers==23.5.26
gast==0.5.4
google-auth==2.26.2
google-auth-oauthlib==1.2.0
google-pasta==0.2.0
grpcio==1.60.0
h5py==3.10.0
idna==3.6
keras==2.15.0
libclang==16.0.6
Markdown==3.5.2
MarkupSafe==2.1.3
ml-dtypes==0.2.0
numpy==1.26.3
oauthlib==3.2.2
opt-einsum==3.3.0
packaging==23.2
pandas==2.1.4
protobuf==4.23.4
pyasn1==0.5.1
pyasn1-modules==0.3.0
python-dateutil==2.8.2
pytz==2023.3.post1
requests==2.31.0
requests-oauthlib==1.3.1
rsa==4.9
scipy==1.11.4
six==1.16.0
tensorboard==2.15.1
tensorboard-data-server==0.7.2
tensorflow==2.15.0
tensorflow-estimator==2.15.0
tensorflow-intel==2.15.0
tensorflow-io-gcs-filesystem==0.31.0
termcolor==2.4.0
typing_extensions==4.9.0
tzdata==2023.1
```

Рисунок 14 – Содержимое файла «requirements.txt».

Формирование файла «environment.yml» (создание списка параметров текущего окружения) с использованием команды «conda env export > environment.yml» (рис. 15) и содержимое этого файла (рис. 16).



```
(Python_laboratory_work_2.14) C:\Users\Andrey\Desktop\пайтон\17 лаба\Python_laboratory_work_2.14\example_environment>conda env export > environment.yml
```

Рисунок 15 – Создание файла «requirements.txt».



Просмотр содержимого файла «environment.yml» (рис. 16).

```
name: Python_laboratory_work_2.14
channels:
  - defaults
dependencies:
  - bzip2=1.0.8=he774522_0
  - ca-certificates=2023.12.12=haa95532_0
  - expat=2.5.0=hd77b12b_0
  - libffi=3.4.4=hd77b12b_0
  - openssl=3.0.12=h2bbff1b_0
  - pip=23.3.1=py310haa95532_0
  - python=3.10.13=he1021f5_0
  - setuptools=68.2.2=py310haa95532_0
  - sqlite=3.41.2=h2bbff1b_0
  - tk=8.6.12=h2bbff1b_0
  - vc=14.2=h21ff451_1
  - vs2015_runtime=14.27.29016=h5e58377_2
  - wheel=0.41.2=py310haa95532_0
  - xz=5.4.5=h8cc25b3_0
  - zlib=1.2.13=h8cc25b3_0
  - pip:
    - absl-py==2.0.0
    - astunparse==1.6.3
    - cachetools==5.3.2
    - certifi==2023.11.17
    - charset-normalizer==3.3.2
    - flatbuffers==23.5.26
    - gast==0.5.4
    - google-auth==2.26.2
    - google-auth-oauthlib==1.2.0
    - google-pasta==0.2.0
    - grpcio==1.60.0
    - h5py==3.10.0
    - idna==3.6
    - keras==2.15.0
```

Рисунок 15 – Содержимое файла «environment.yml».

#### Ответы на контрольные вопросы:

1. Менеджер пакетов «pip» позволяет устанавливать пакеты, не входящие в стандартную библиотеку.
2. Зачастую менеджер пакетов «pip» устанавливается по умолчанию, однако если этого не произошло, то можно использовать команды его установки: в Anaconda – «conda install pip». В Windows для этого нужно скачать установочный скрипт «get-pip.py», и открыть его в командной строке при помощи команды «python get-pip.py».
3. Менеджер пакетов «pip» находит требуемые пакеты в репозитории Python Package Index (PyPI).
4. Для установки пакетов при помощи «pip» существует команда «pip install <название\_пакета>».

5. Если необходимо установить не последнюю версию пакета, то к стандартной команде установки «`pip install <название_пакета>`» добавляется «`==<требуемая_версия>`».

6. Для установки пакета из любого «git» репозитория при помощи «pip» можно воспользоваться командой «`pip install git+<ссылка_на_требуемый_репозиторий>`».

7. «pip» позволяет устанавливать пакеты из локальной директории, если использовать команду «`pip install <полный_путь_к_пакету>`».

8. Команда «`pip uninstall <название_пакета>`» позволяет избавиться от пакета указанного пакета.

9. Команда «`pip install --upgrade <название_пакета>`» позволяет обновить требуемый пакет до последней версии.

10. Все установленные пакеты можно увидеть при помощи команды «`pip list`».

11. Виртуальные окружения позволяют изолировать проекты и их зависимости. Появилась такая необходимость из-за ограничения Python – может быть установлено не больше 1 версии пакета.

12. Основные этапы работы с виртуальным окружением: создание окружения («`conda create -n <название_окружения>`»), активация окружения («`conda activate <название_окружения>`»), деактивация окружения («`conda deactivate <название_окружения>`») и удаление окружения («`conda env remove -name <название_окружения>`»).

13. При работе с «venv» необходимо пользоваться командами: создание окружения («`python -m venv <название_окружения>`»), активация («`source <название_окружения> /bin/activate`»), деактивация («`deactivate`»).

14. При работе с «virtualenv» необходимо пользоваться командами: создание окружения («`virtualenv <название_окружения>`»), активация («`source <название_окружения> /bin/activate`»), деактивация («`deactivate`»).

15. При работе с «virtualenv» необходимо пользоваться командами: создание окружения «`pipenv install`», активация «`pipenv shell`», деактивация «`exit`» и удаление «`pipenv -remove`».

16. Файл «`requirements.txt`» является списком зависимостей проекта. Для его создания необходимо использовать команду «`pip freeze > requirements.txt`». Зависимости представлены в виде: «пакет»==«версия».

17. Менеджер пакетов «conda», в отличие от «pip», способен управлять библиотеками и окружением (даже бинарными зависимостями). «Pip» же способен управлять только пакетами.

18. Менеджер пакетов «conda» входит в дистрибутивы Anaconda и Miniconda.

19. Команда «`conda create -n <название_окружения>`» позволяет создать новое окружение при помощи «conda».

20. Команда «`conda activate <название_окружения>`» позволяет активировать требуемое окружение, а команда «`conda install <название_пакета>`» позволяет устанавливать требуемые пакеты.

21. Команда «`conda deactivate <название_окружения>`» позволяет деактивировать текущее окружение, а команда «`conda env remove --name <название_окружения>`» удаляет требуемое окружение.

22. Файл «`environment.yml`» хранит в себе параметры окружения, благодаря которым можно соответствующее окружение легко воссоздать. Команда «`conda env export > environment.yml`» создаёт такой файл относительно активного окружения (перенаправляет параметры окружения в файл и создает его).

23. Для создания окружения на основе параметров из файла «`environment.yml`», можно воспользоваться командой «`conda env create -f environment.yml`».

24. Если установлены «conda» и «PyCharm», то в настройках «PyCharm» можно в разделе «Интерпретаторы» необходимо добавить новый интерпретатор «conda». После успешной настройки нового интерпретатора

появляется возможность работы с виртуальным окружением, установки пакетов.

25. Так как файлы «requirements.txt» и «environment.yml» содержат информацию о зависимостях проекта, то их наличие в git облегчает воссоздание того же окружения на других системах и для других разработчиков.

Вывод: в ходе выполнения лабораторной работы была подробно рассмотрена работа с окружениями для Python, а также работа с пакетами. Были приобретены навыки по работе с менеджерами пакетов «pip» и «conda». Были установлены различные пакеты для проекта с помощью разных пакетов. Также были изучены способы получения списка зависимостей проекта и параметры окружения.