

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

(подпись)

Руководитель практики:
Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Ход работы

1. Изучаем теоретический материал работы

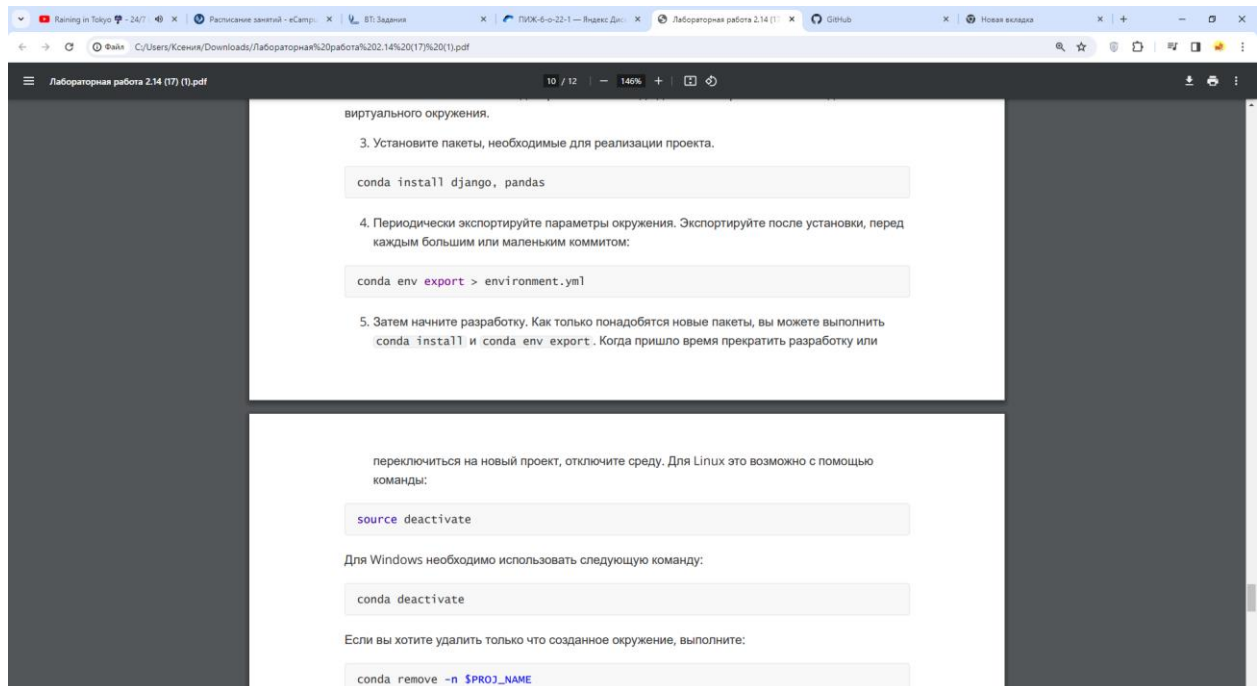


Рисунок 1.1 – Изучение материала для лабораторной работы

2. Создаем общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и язык программирования Python

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

☒ 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

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

Choose a license

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

This will set `main` as the default branch. Change the default name in your settings.

☐ You are creating a public repository in your personal account.

Рисунок 2.1 – Настройка репозитория

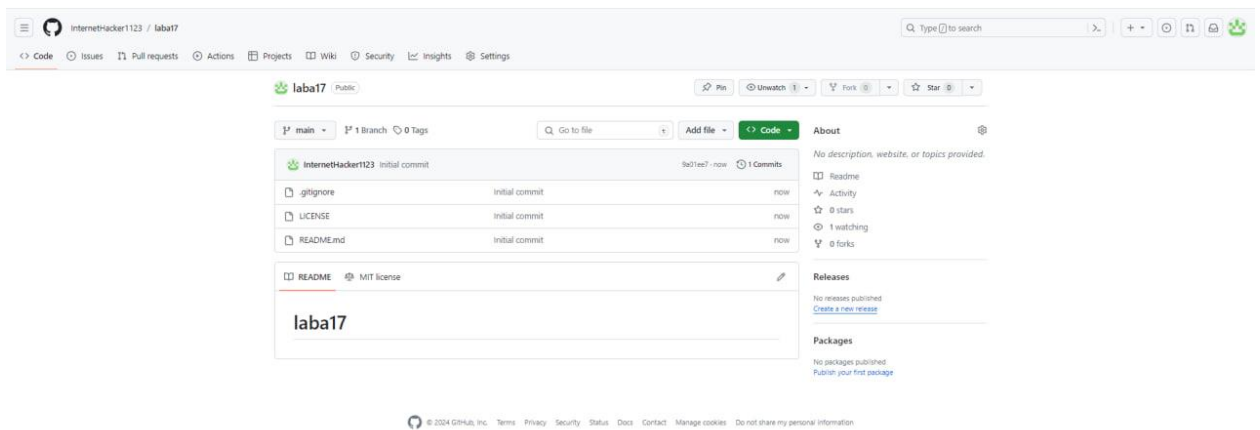


Рисунок 2.2 – Готовый репозиторий

3. Выполняем клонирование созданного репозитория

```
C:\Windows\system32\cmd.exe

C:\Users\tyt\Desktop\SE>mkdir laba17
C:\Users\tyt\Desktop\SE>cd laba17
C:\Users\tyt\Desktop\SE\laba17>git clone https://github.com/InternetHacker1123/laba17.git
Cloning into 'laba17'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
C:\Users\tyt\Desktop\SE\laba17>
```

Рисунок 3.1 – Клонирование репозитория на локальный диск

4. Организовываем свой репозиторий в соответствии с моделью ветвления git-flow

```
C:\Windows\system32\cmd.exe

C:\Users\tyt\Desktop\SE\laba17\laba17>git branch develop
C:\Users\tyt\Desktop\SE\laba17\laba17>git checkout develop
Switched to branch 'develop'

C:\Users\tyt\Desktop\SE\laba17\laba17>git push origin develop
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/InternetHacker1123/laba17/pull/new/develop
remote:
To https://github.com/InternetHacker1123/laba17.git
 * [new branch]      develop -> develop

C:\Users\tyt\Desktop\SE\laba17\laba17>git status
On branch develop
nothing to commit, working tree clean

C:\Users\tyt\Desktop\SE\laba17\laba17>_
```

Рисунок 5.1 – Создание ветки develop от ветки main

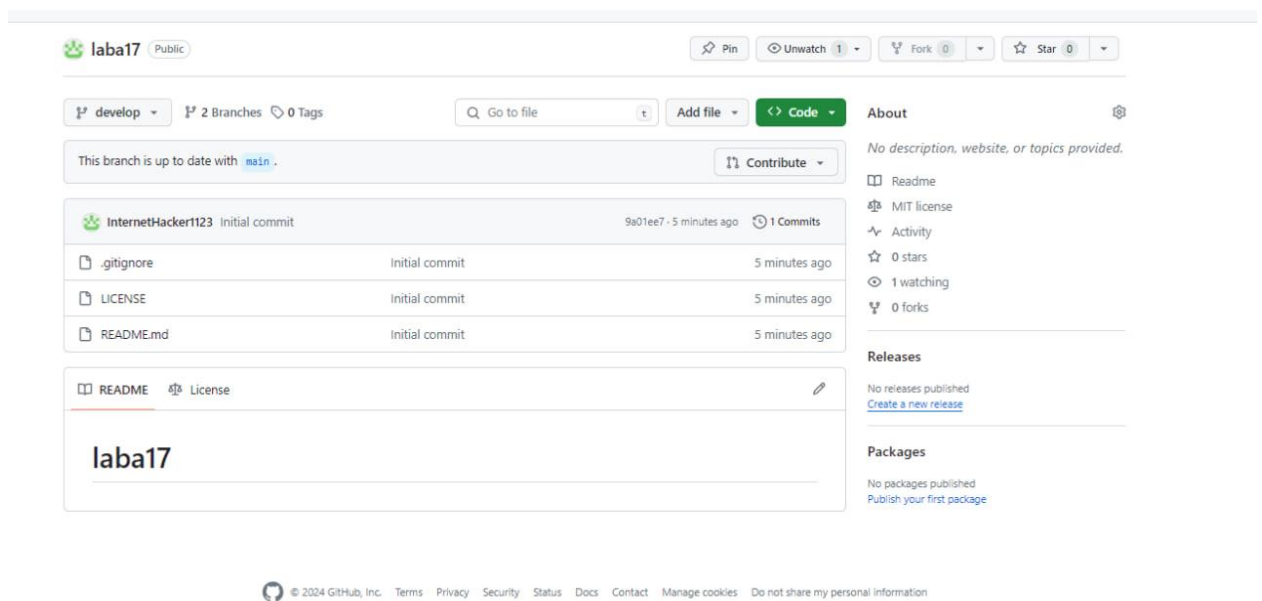


Рисунок 5.2 – Ветка develop на GitHub

5. Создаем виртуальное окружение Anaconda с именем репозитория.

```
Администратор: Windows PowerShell
(base) PS C:\Users\tyt\Desktop\SE\laba17\laba17> mkdir project

Каталог: C:\Users\tyt\Desktop\SE\laba17\laba17

Mode                LastWriteTime         Length Name
----                -
d-----          12.02.2024    20:05         project

(base) PS C:\Users\tyt\Desktop\SE\laba17\laba17> cd .\project\
(base) PS C:\Users\tyt\Desktop\SE\laba17\laba17\project> copy NUL > main.py
copy : Не удается найти путь "C:\Users\tyt\Desktop\SE\laba17\laba17\project\NUL", так как он не существует.
строка:1 знак:1
+ ~~~~~
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (C:\Users\tyt\De...a17\project\NUL:String) [Copy-Item], ItemNotFoundExce
  ption
+ FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.CopyItemCommand

(base) PS C:\Users\tyt\Desktop\SE\laba17\laba17\project>
```

Рисунок 5.1 – Создание проекта

```
Администратор: Windows PowerShell

package | build
-----|-----
ca-certificates-2023.12.12 | haa95532_0 | 127 KB
certifi-2022.12.7 | py37haa95532_0 | 149 KB
openssl-1.1.1w | h2bbff1b_0 | 5.5 MB
pip-22.3.1 | py37haa95532_0 | 2.7 MB
python-3.7.16 | h6244533_0 | 17.2 MB
setuptools-65.6.3 | py37haa95532_0 | 1.1 MB
sqlite-3.41.2 | h2bbff1b_0 | 894 KB
vc-14.2 | h21ff451_1 | 8 KB
vs2015_runtime-14.27.29016 | h5e58377_2 | 1007 KB
wheel-0.38.4 | py37haa95532_0 | 82 KB
wincertstore-0.2 | py37haa95532_2 | 15 KB
-----|-----
Total: | 28.8 MB

The following NEW packages will be INSTALLED:

ca-certificates pkgs/main/win-64::ca-certificates-2023.12.12-haa95532_0
certifi pkgs/main/win-64::certifi-2022.12.7-py37haa95532_0
openssl pkgs/main/win-64::openssl-1.1.1w-h2bbff1b_0
pip pkgs/main/win-64::pip-22.3.1-py37haa95532_0
python pkgs/main/win-64::python-3.7.16-h6244533_0
setuptools pkgs/main/win-64::setuptools-65.6.3-py37haa95532_0
sqlite pkgs/main/win-64::sqlite-3.41.2-h2bbff1b_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-0.38.4-py37haa95532_0
wincertstore pkgs/main/win-64::wincertstore-0.2-py37haa95532_2

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Executing transaction: done
#
# To activate this environment, use
#
# $ conda activate project
#
# To deactivate an active environment, use
#
# $ conda deactivate

(base) PS C:\Users\tyt\Desktop\SE\laba17\laba17\project> conda activate project_
(project) PS C:\Users\tyt\Desktop\SE\laba17\laba17\project>
```

Рисунок 5.2 – Создание виртуального окружения

6. Устанавливаем в виртуальное окружение следующие пакеты: `pip`, `NumPy`, `Pandas`, `SciPy`.

```
Администратор: Windows PowerShell

package                                     build
-----
blas-1.0                                   mkl 6 KB
bottleneck-1.3.5                          py37h080aedc_0 105 KB
fftw-3.3.9                                h2bbff1b_1 672 KB
icc_rt-2022.1.0                           h6049295_2 6.5 MB
intel-openmp-2021.4.0                     haa95532_3556 2.2 MB
mkl-2021.4.0                              haa95532_640 114.9 MB
mkl-service-2.4.0                         py37h2bbff1b_0 49 KB
mkl_fft-1.3.1                             py37h277e83a_0 135 KB
mkl_random-1.2.2                          py37hf11a4ad_0 216 KB
numexpr-2.8.4                             py37h5b0cc5e_0 127 KB
numpy-1.21.5                              py37h7a0a035_3 25 KB
numpy-base-1.21.5                        py37hca35cd5_3 4.4 MB
packaging-22.0                            py37haa95532_0 67 KB
pandas-1.3.5                              py37h6214cd6_0 8.4 MB
python-dateutil-2.8.2                    pyhd3eb1b0_0 233 KB
pytz-2022.7                               py37haa95532_0 210 KB
scipy-1.7.3                               py37h7a0a035_2 13.8 MB
six-1.16.0                                pyhd3eb1b0_1 18 KB
Total: 151.9 MB

The following NEW packages will be INSTALLED:

blas                pkgs/main/win-64::blas-1.0-mkl
bottleneck          pkgs/main/win-64::bottleneck-1.3.5-py37h080aedc_0
fftw                pkgs/main/win-64::fftw-3.3.9-h2bbff1b_1
icc_rt              pkgs/main/win-64::icc_rt-2022.1.0-h6049295_2
intel-openmp        pkgs/main/win-64::intel-openmp-2021.4.0-haa95532_3556
mkl                 pkgs/main/win-64::mkl-2021.4.0-haa95532_640
mkl-service         pkgs/main/win-64::mkl-service-2.4.0-py37h2bbff1b_0
mkl_fft             pkgs/main/win-64::mkl_fft-1.3.1-py37h277e83a_0
mkl_random          pkgs/main/win-64::mkl_random-1.2.2-py37hf11a4ad_0
numexpr             pkgs/main/win-64::numexpr-2.8.4-py37h5b0cc5e_0
numpy               pkgs/main/win-64::numpy-1.21.5-py37h7a0a035_3
numpy-base         pkgs/main/win-64::numpy-base-1.21.5-py37hca35cd5_3
packaging           pkgs/main/win-64::packaging-22.0-py37haa95532_0
pandas              pkgs/main/win-64::pandas-1.3.5-py37h6214cd6_0
python-dateutil     pkgs/main/noarch::python-dateutil-2.8.2-pyhd3eb1b0_0
pytz                pkgs/main/win-64::pytz-2022.7-py37haa95532_0
scipy               pkgs/main/win-64::scipy-1.7.3-py37h7a0a035_2
six                 pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1

Proceed ([y]/n)? y

Downloading and Extracting Packages
```

Рисунок 7.1 – Установка пакетов

7. Попробуем установить менеджером пакетов conda пакет TensorFlow. Ошибка возникла, но оказалось решаемой, с помощью команды conda-forge.

```
Администратор: Windows PowerShell

zipp-3.11.0                                py37haa95532_0 20 KB
zlib-1.2.13                               h8cc25b3_0 113 KB
Total: 87.0 MB

The following NEW packages will be INSTALLED:

abs1-py                conda-forge/noarch::abs1-py-2.1.0-pyhd8ed1ab_0
astor                  conda-forge/noarch::astor-0.8.1-pyh9f0ad1d_0
ast                    conda-forge/noarch::ast-0.5.4-pyhd8ed1ab_0
google-pasta           conda-forge/noarch::google-pasta-0.2.0-pyh8c360ce_0
grpcio                 pkgs/main/win-64::grpcio-1.42.0-py37hc60d5dd_0
h5py                   conda-forge/win-64::h5py-2.10.0-nompi-py37he280515_106
hdf5                   conda-forge/win-64::hdf5-1.10.6-nompi1-he0bbb20_101
importlib-metadata     conda-forge/win-64::importlib-metadata-4.11.4-py37h03978a9_0
keras-applications     conda-forge/noarch::keras-applications-1.0.8-py_1
keras-preprocessing   conda-forge/noarch::keras-preprocessing-1.1.2-pyhd8ed1ab_0
libprotobuf            pkgs/main/win-64::libprotobuf-3.20.3-h23ce68f_0
markdown               conda-forge/noarch::markdown-3.5.2-pyhd8ed1ab_0
protobuf               pkgs/main/win-64::protobuf-3.20.3-py37hd77b12b_0
pyreadline             conda-forge/win-64::pyreadline-2.1-py37h03978a9_1006
python_abi             conda-forge/win-64::python_abi-3.7-2_cp37m
tensorboard            conda-forge/win-64::tensorboard-1.14.0-py37_0
tensorflow              conda-forge/win-64::tensorflow-1.14.0-h1f41ff6_0
tensorflow-base        conda-forge/win-64::tensorflow-base-1.14.0-py37hc8dfbb8_0
tensorflow-estimator   conda-forge/win-64::tensorflow-estimator-1.14.0-py37h5cald4c_0
termcolor              conda-forge/noarch::termcolor-2.2.0-pyhd8ed1ab_0
typing_extensions      conda-forge/noarch::typing_extensions-4.5.0-pyha770c72_0
werkzeug               conda-forge/noarch::werkzeug-2.1.2-pyhd8ed1ab_1
wrapt                  pkgs/main/win-64::wrapt-1.14.1-py37h2bbff1b_0
zipp                   pkgs/main/win-64::zipp-3.11.0-py37haa95532_0
zlib                   pkgs/main/win-64::zlib-1.2.13-h8cc25b3_0

The following packages will be UPDATED:

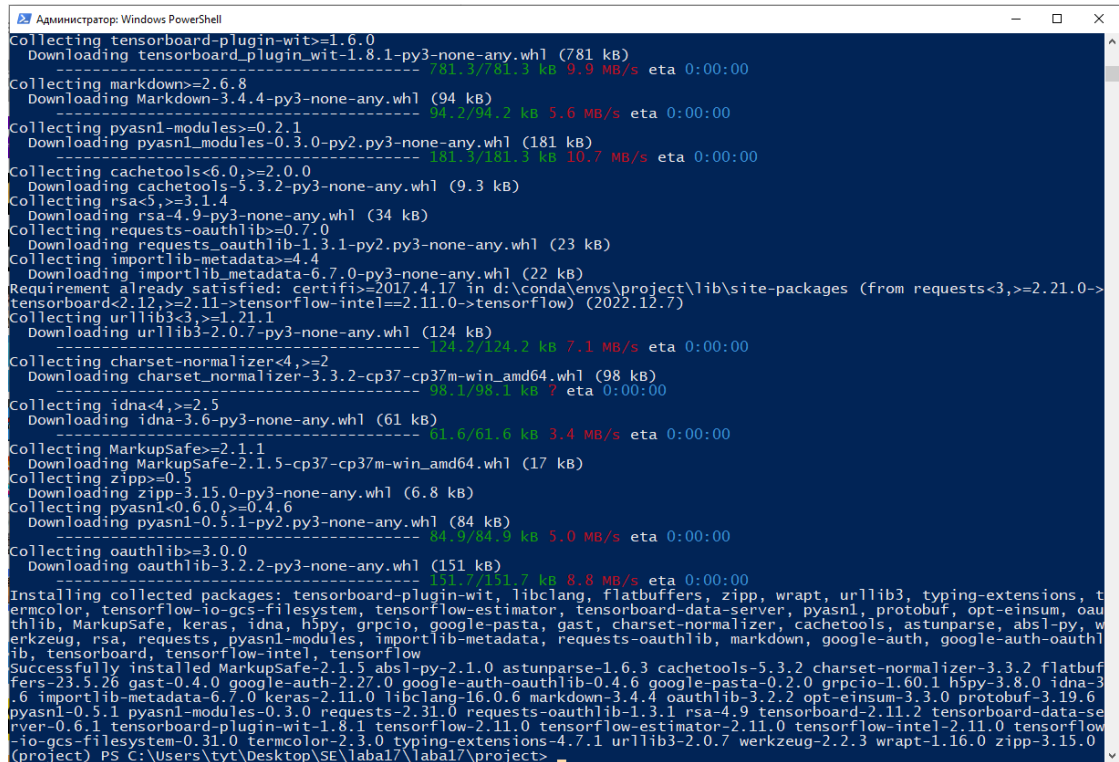
ca-certificates        pkgs/main::ca-certificates-2023.12.12~ --> conda-forge::ca-certificates-2024.2.2-h56e8100_0
certifi                pkgs/main/win-64::certifi-2022.12.7-p~ --> conda-forge/noarch::certifi-2024.2.2-pyhd8ed1ab_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
Preparing transaction: done
Executing transaction: done
(project) PS C:\Users\tyt\Desktop\SE\lab17\lab17\project>
```

Рисунок 8.1 – Установка tensorflow через conda

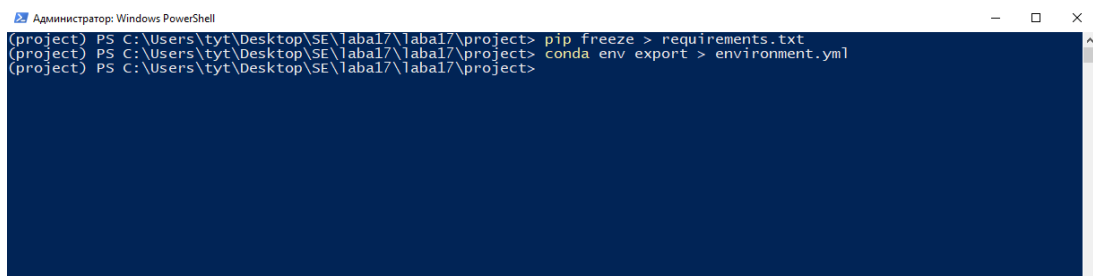
8. Пробуем установить пакет TensorFlow с помощью менеджера пакетов `pip`.



```
Администратор: Windows PowerShell
Collecting tensorboard-plugin-wit==1.6.0
  Downloading tensorboard_plugin_wit-1.8.1-py3-none-any.whl (781 kB)
-----
781.3/781.3 kB 9.9 MB/s eta 0:00:00
Collecting markdown>=2.6.8
  Downloading Markdown-3.4.4-py3-none-any.whl (94 kB)
-----
94.2/94.2 kB 5.6 MB/s eta 0:00:00
Collecting pyasn1-modules>=0.2.1
  Downloading pyasn1_modules-0.3.0-py2.py3-none-any.whl (181 kB)
-----
181.3/181.3 kB 10.7 MB/s eta 0:00:00
Collecting cachetools<6.0,>=2.0.0
  Downloading cachetools-5.3.2-py3-none-any.whl (9.3 kB)
Collecting rsa<5,>=3.1.4
  Downloading rsa-4.9-py3-none-any.whl (34 kB)
Collecting requests-oauthlib>=0.7.0
  Downloading requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)
Collecting importlib-metadata>=4.4
  Downloading importlib_metadata-6.7.0-py3-none-any.whl (22 kB)
Requirement already satisfied: certifi>=2017.4.17 in d:\conda\envs\project\lib\site-packages (from requests<3,>=2.21.0->
tensorboard<2.12,>=2.11->tensorflow-intel==2.11.0->tensorflow) (2022.12.7)
Collecting urllib3<3,>=1.21.1
  Downloading urllib3-2.0.7-py3-none-any.whl (124 kB)
-----
124.2/124.2 kB 7.1 MB/s eta 0:00:00
Collecting charset-normalizer<4,>=2
  Downloading charset_normalizer-3.3.2-cp37-cp37m-win_amd64.whl (98 kB)
-----
98.1/98.1 kB ? eta 0:00:00
Collecting idna<4,>=2.5
  Downloading idna-3.6-py3-none-any.whl (61 kB)
-----
61.6/61.6 kB 3.4 MB/s eta 0:00:00
Collecting MarkupSafe>=2.1.1
  Downloading MarkupSafe-2.1.5-cp37-cp37m-win_amd64.whl (17 kB)
Collecting zipp>=0.5
  Downloading zipp-3.15.0-py3-none-any.whl (6.8 kB)
Collecting pyasn1<0.6.0,>=0.4.6
  Downloading pyasn1-0.5.1-py2.py3-none-any.whl (84 kB)
-----
84.9/84.9 kB 5.0 MB/s eta 0:00:00
Collecting oauthlib>=3.0.0
  Downloading oauthlib-3.2.2-py3-none-any.whl (151 kB)
-----
151.7/151.7 kB 8.8 MB/s eta 0:00:00
Installing collected packages: tensorboard-plugin-wit, libclang, flatbuffers, zipp, wrapt, urllib3, typing-extensions, t
ermcolor, tensorflow-io-gcs-fsfilesystem, tensorflow-estimator, tensorboard-data-server, pyasn1, protobuf, opt-einsum, oau
thlib, MarkupSafe, keras, idna, h5py, grpcio, google-pasta, gast, charset-normalizer, cachetools, astunparse, absl-py, w
erkzeug, rsa, requests, pyasn1-modules, importlib-metadata, requests-oauthlib, markdown, google-auth, google-auth-oauthl
ib, tensorboard, tensorflow-intel, tensorflow
Successfully installed MarkupSafe-2.1.5 absl-py-2.1.0 astunparse-1.6.3 cachetools-5.3.2 charset-normalizer-3.3.2 flatbuf
fers-23.5.26 gast-0.4.0 google-auth-2.27.0 google-auth-oauthlib-0.4.6 google-pasta-0.2.0 grpcio-1.60.1 h5py-3.8.0 idna-3
.6 importlib-metadata-6.7.0 keras-2.11.0 libclang-16.0.6 markdown-3.4.4 oauthlib-3.2.2 opt-einsum-3.3.0 protobuf-3.19.6
pyasn1-0.5.1 pyasn1-modules-0.3.0 requests-2.31.0 requests-oauthlib-1.3.1 rsa-4.9 tensorboard-2.11.2 tensorboard-data-se
rver-0.6.1 tensorboard-plugin-wit-1.8.1 tensorflow-2.11.0 tensorflow-estimator-2.11.0 tensorflow-intel-2.11.0 tensorflow
-io-gcs-fsfilesystem-0.31.0 termcolor-2.3.0 typing-extensions-4.7.1 urllib3-2.0.7 werkzeug-2.2.3 wrapt-1.16.0 zipp-3.15.0
(project) PS C:\Users\tyt\Desktop\SE\lab17\lab17\project>
```

Рисунок 8.1 – Установка tensorflow через `pip`

9. Сформируем файлы `requirements.txt` и `environment.yml`.
Анализируем содержимое этих файлов.



```
Администратор: Windows PowerShell
(project) PS C:\Users\tyt\Desktop\SE\lab17\lab17\project> pip freeze > requirements.txt
(project) PS C:\Users\tyt\Desktop\SE\lab17\lab17\project> conda env export > environment.yml
(project) PS C:\Users\tyt\Desktop\SE\lab17\lab17\project>
```

Рисунок 9.1 – Формирование файлов зависимостей

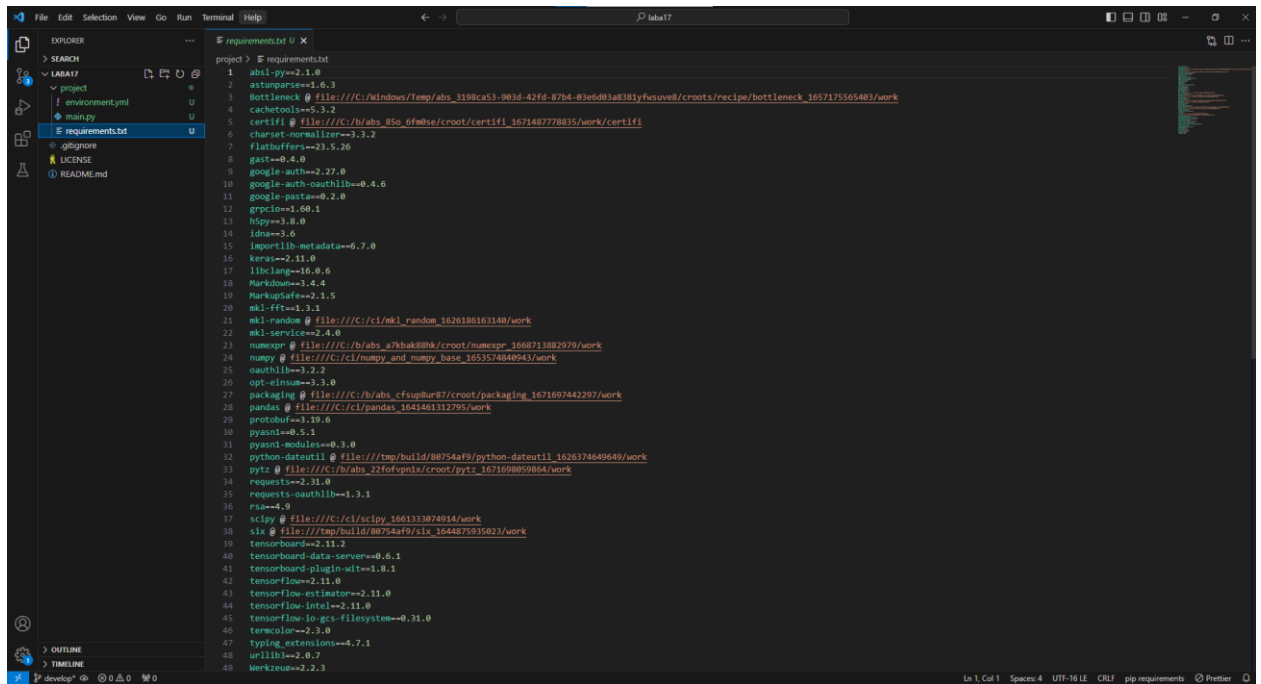


Рисунок 9.2 – Файл requirements.txt

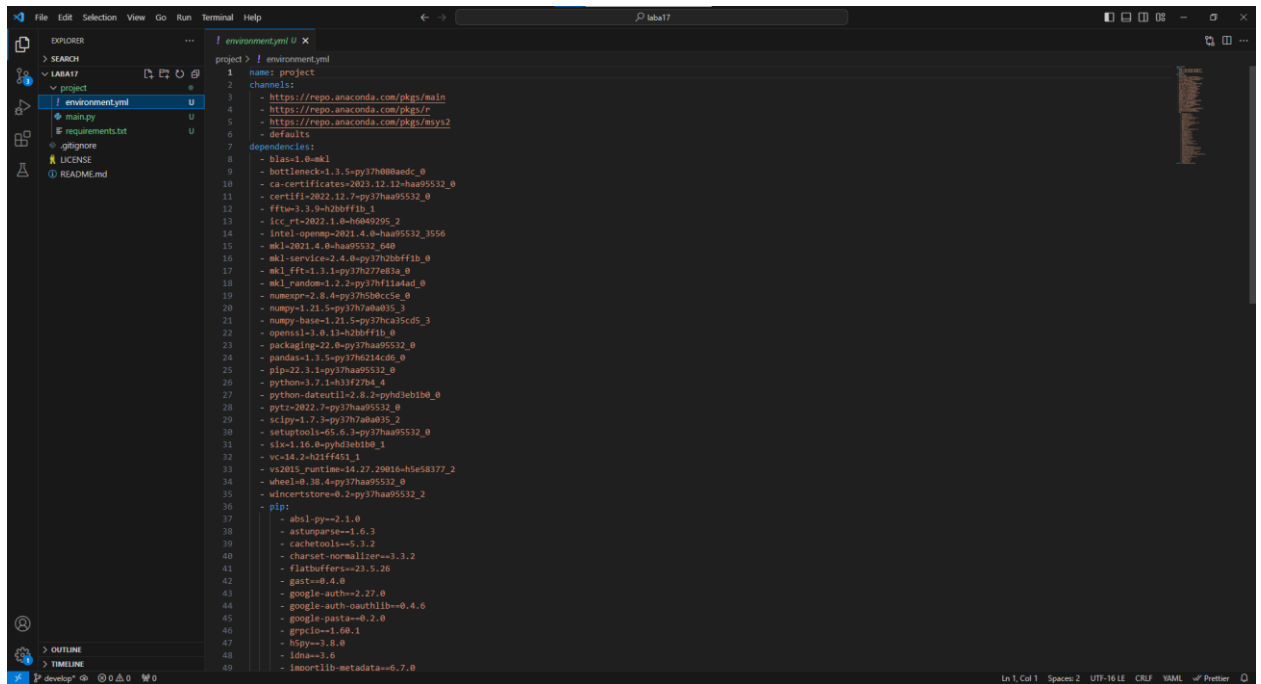


Рисунок 9.3 – Файл environment.yml

10. Фиксируем сделанные изменения в репозитории.

```
C:\Users\tyt\Desktop\SE\laba17\laba17>git push --set-upstream origin develop
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 2.56 KiB | 2.56 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/InternetHacker1123/laba17.git
   9a01ee7..0f86ac0  develop -> develop
branch 'develop' set up to track 'origin/develop'.

C:\Users\tyt\Desktop\SE\laba17\laba17>
```

Рисунок 10.1 – Коммит файлов в репозитории git

Контрольные вопросы

1. Каким способом можно установить пакет Python, не входящий в стандартную библиотеку?

Для скачивания и установки используется специальная утилита, которая называется `pip`.

2. Как осуществить установку менеджера пакетов `pip`?

`pip install ProjectName`

3. Откуда менеджер пакетов `pip` по умолчанию устанавливает пакеты?

Пакеты, установленные с помощью `pip`, обычно устанавливаются в директории, где установлен Python.

4. Как установить последнюю версию пакета с помощью `pip`?

`pip install ProjectName`

5. Как установить заданную версию пакета с помощью `pip`?

`pip install ProjectName==3.2`

6. Как установить пакет из git репозитория (в том числе GitHub) с помощью `pip`?

`pip install -e git+https://gitrepo.com/ProjectName.git`

7. Как установить пакет из локальной директории с помощью pip?

```
pip install ./dist/ProjectName.tar.gz
```

8. Как удалить установленный пакет с помощью pip?

```
pip uninstall ProjectName
```

9. Как обновить установленный пакет с помощью pip?

```
pip install --upgrade ProjectName
```

10. Как отобразить список установленных пакетов с помощью pip?

```
pip list
```

11. Каковы причины появления виртуальных окружений в языке Python?

Виртуальное окружение позволяет создавать изолированные среды для каждого проекта, в которых можно устанавливать и управлять зависимостями и пакетами, не вмешиваясь в другие проекты или системные установки Python.

12. Каковы основные этапы работы с виртуальными окружениями?

Основные этапы:

1. Установка инструментов;
2. Создание виртуального окружения;
3. Активация виртуального окружения;
4. Установка пакетов и зависимостей;
5. Деактивация виртуального окружения

13. Как осуществляется работа с виртуальными окружениями с помощью venv?

Для создания виртуального окружения достаточно дать команду в формате: `python3 -m venv <путь к папке виртуального окружения>`. Чтобы активировать окружение под нам нужно дать команду: `> env\\Scripts\\activate`. Чтобы переключиться с одного окружения на другое нам нужно выполнить команду деактивации и команду активации другого виртуального окружения:

```
deactivate
```

14. Как осуществляется работа с виртуальными окружениями с помощью virtualenv?

Для начала пакет нужно установить. Установку можно выполнить командой: `python3 -m pip install virtualenv`. Активация и деактивация такая же, как у стандартной утилиты Python

15. Изучите работу с виртуальными окружениями pipenv. Как осуществляется работа с виртуальными окружениями pipenv?

Работа с виртуальными окружениями pipenv:

1. Установка pipenv, если его нет, используя команду: `pip install pipenv`

2. Создание новое виртуальное окружение, в котором будут установлены все зависимости проекта, с помощью команды: `pipenv install`

3. Активация виртуального окружение, используя команду: `pipenv shell`

4. Установка необходимых пакетов в виртуальное окружение, например: `pipenv install requests`

5. После завершения работы с виртуальным окружением деактивация с помощью команды: `exit`

16. Каково назначение файла requirements.txt? Как создать этот файл? Какой он имеет формат?

Файл requirements.txt используется для хранения списка всех зависимостей вашего проекта. Этот файл позволяет другим разработчикам или системам легко установить все необходимые зависимости для вашего проекта. Что бы его сохранить, нужно перенаправить вывод команды в файл: `pip freeze > requirements.txt`. Данный формат является обычным текстовым файлом, где указано название пакета python, его версия и условие, больше, меньше, равно.

17. В чем преимущества пакетного менеджера conda по сравнению с пакетным менеджером pip?

Основная проблема заключается в том, что pip, easy_install и virtualenv ориентированы на Python. Эти инструменты игнорируют библиотеки

зависимостей, реализованные с использованием других языков. Например, XSLT, HDF5, MKL и другие, которые не имеют setup.py в исходном коде и не устанавливают файлы в директорию site-packages. Conda же способна управлять пакетами как для Python, так и для C/ C++, R, Ruby, Lua, Scala и других. Conda устанавливает двоичные файлы, поэтому работу по компиляции пакета самостоятельно выполнять не требуется (по сравнению с pip).

18. В какие дистрибутивы Python входит пакетный менеджер conda?

Пакетный менеджер conda входит в дистрибутив Anaconda, Miniconda и другие связанные с ними дистрибутивы Python

19. Как создать виртуальное окружение conda?

Начиная проект, создайте чистую директорию и дайте ей понятное короткое имя. Для Windows это будет соответствовать набору команд:

```
mkdir %PROJ_NAME%
```

```
cd %PROJ_NAME%
```

```
copy NUL > main.py
```

20. Как активировать и установить пакеты в виртуальное окружение conda?

```
conda create -n %PROJ_NAME% python=3.7 conda
```

```
activate %PROJ_NAME%
```

```
conda install django, pandas
```

21. Как деактивировать и удалить виртуальное окружение conda?

```
conda deactivate
```

```
conda remove -n $PROJ_NAME
```

22. Каково назначение файла environment.yml? Как создать этот файл?

Файл environment.yml в среде conda используется для описания зависимостей вашего проекта, включая список пакетов и их версии. Этот файл может быть использован для воссоздания точной среды с необходимыми пакетами.

23. Как создать виртуальное окружение conda с помощью файла environment.yml?

Файл `environment.yml` позволит воссоздать окружение в любой нужный момент. Достаточно набрать:

```
conda env create -f environment.yml
```

24. Самостоятельно изучите средства IDE PyCharm для работы с виртуальными окружениями conda. Опишите порядок работы с виртуальными окружениями conda в IDE PyCharm.

Для работы с виртуальными окружениями conda в IDE PyCharm, следуем этим шагам:

1. Открытие проекта: Откройте свой проект в IDE PyCharm.
2. Настройка интерпретатора
3. Выбор интерпретатора
4. Создание виртуального окружения: В появившемся окне выберите "Conda Environment" слева и затем "Existing environment". Укажите путь к каталогу, где хранится ваше виртуальное окружение conda, или создайте новое виртуальное окружение, указав путь к каталогу Miniconda или Anaconda.
5. Применение изменений
6. Выбор виртуального окружения
7. Работа с виртуальным окружением

Теперь вы можете легко работать с виртуальными окружениями conda прямо из IDE PyCharm, что поможет вам эффективно управлять зависимостями и пакетами в вашем проекте.

25. Почему файлы `requirements.txt` и `environment.yml` должны храниться в репозитории git?

Хранение файлов `requirements.txt` и `environment.yml` в репозитории Git является хорошей практикой, которая облегчает совместную работу над проектом, обеспечивает воспроизводимость окружения и упрощает управление зависимостями.