МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет по лабораторной работе № 1
«Модули и пакеты»

по дисциплине «Основы программной инженерии»

Выполнила:
Коновалова Виктория
Николаевна
2 курс, ПИЖ-б-о-21-1
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2023 г.

ВЫПОЛНЕНИЕ

1. Изучить теоретический материал работы.

2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия

MIT и язык программирования Python.

3. Выполните клонирование созданного репозитория.

4. Организуйте свой ре+позиторий в соответствие с моделью ветвления git-flow.

5. Создайте виртуальное окружение Anaconda с именем репозитория.

```
C:\Users\vika1>cd "C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи"

C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи>python -m venv env
```

Рисунок 2 – Создание виртуального окружения

```
C:\Users\vika1>pip --version
pip 22.2.2 from C:\Users\vika1\anaconda3\lib\site-packages\pip (python 3.9)
```

Рисунок 1 – Проверка на наличие pip

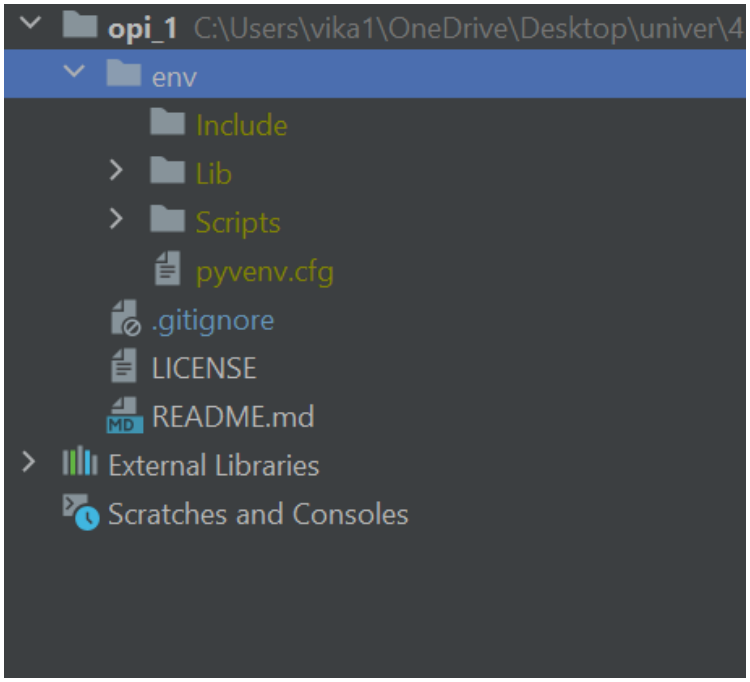

Рисунок 3 – Созданная папка окружения

```
C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1>cd env

C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\env>cd Scripts

C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\env\Scripts>activate.bat

(env) C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\env\Scripts>
```

Рисунок 4 – Активация виртуального окружения

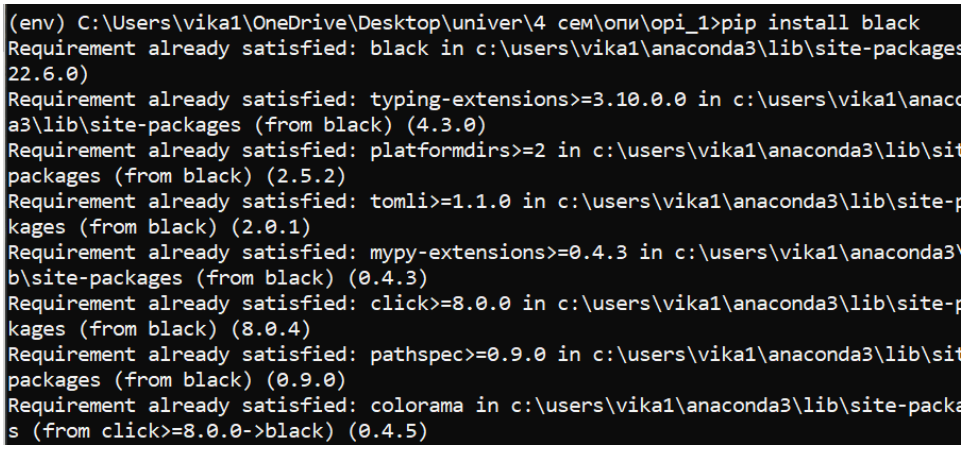

Рисунок 5 – Установка пакета black

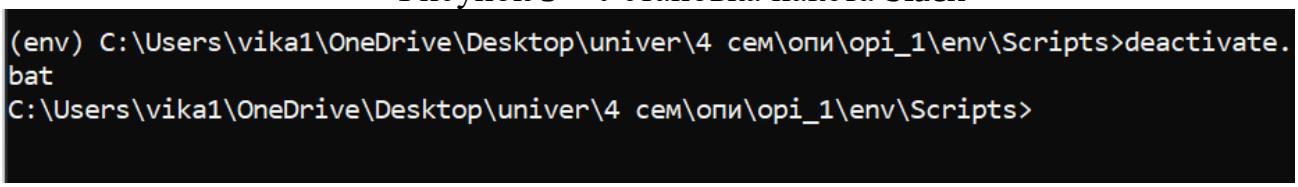

Рисунок 6 – Деактивицизация виртуального окружения

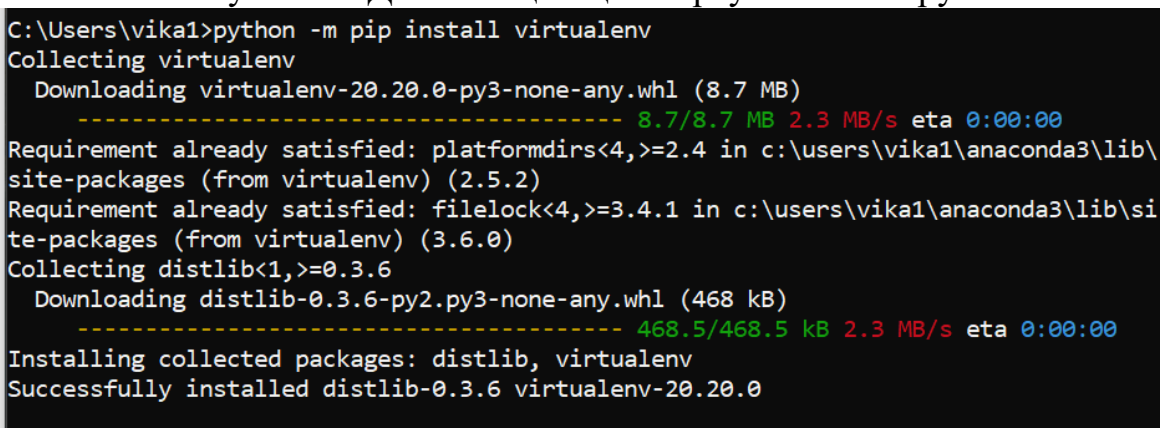

Рисунок 7 – Установка виртуального окружения

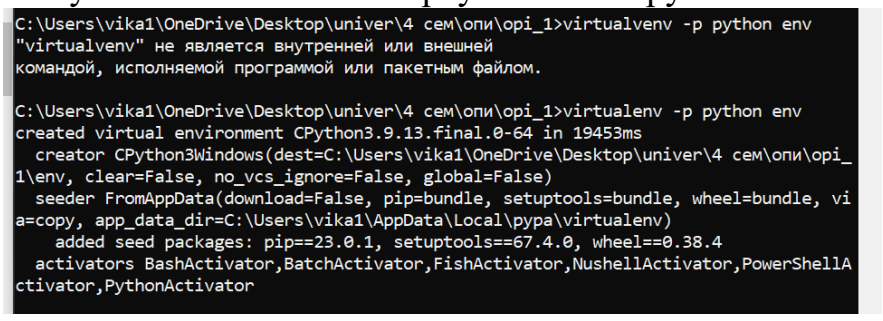

Рисунок 7 – Создание виртуального окружения

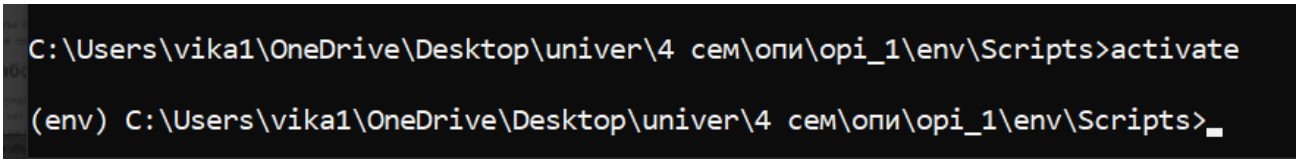

Рисунок 8 – Активизация виртуального окружения

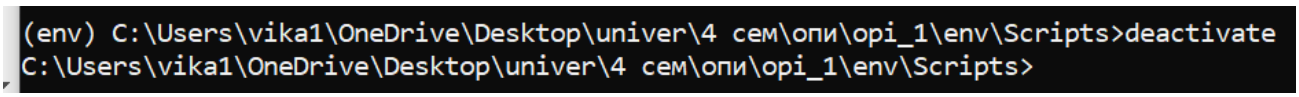

Рисунок 9– Деактивицазия виртуального окружения

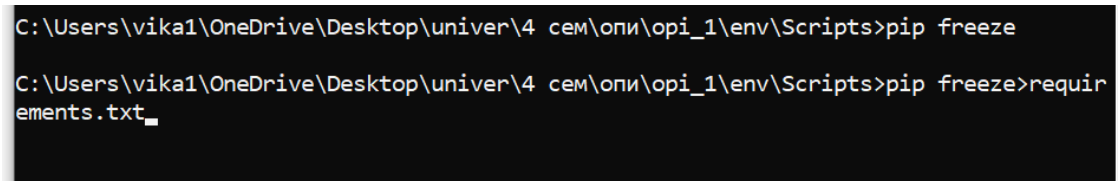

Рисунок 10 – Список пакетных зависимостей

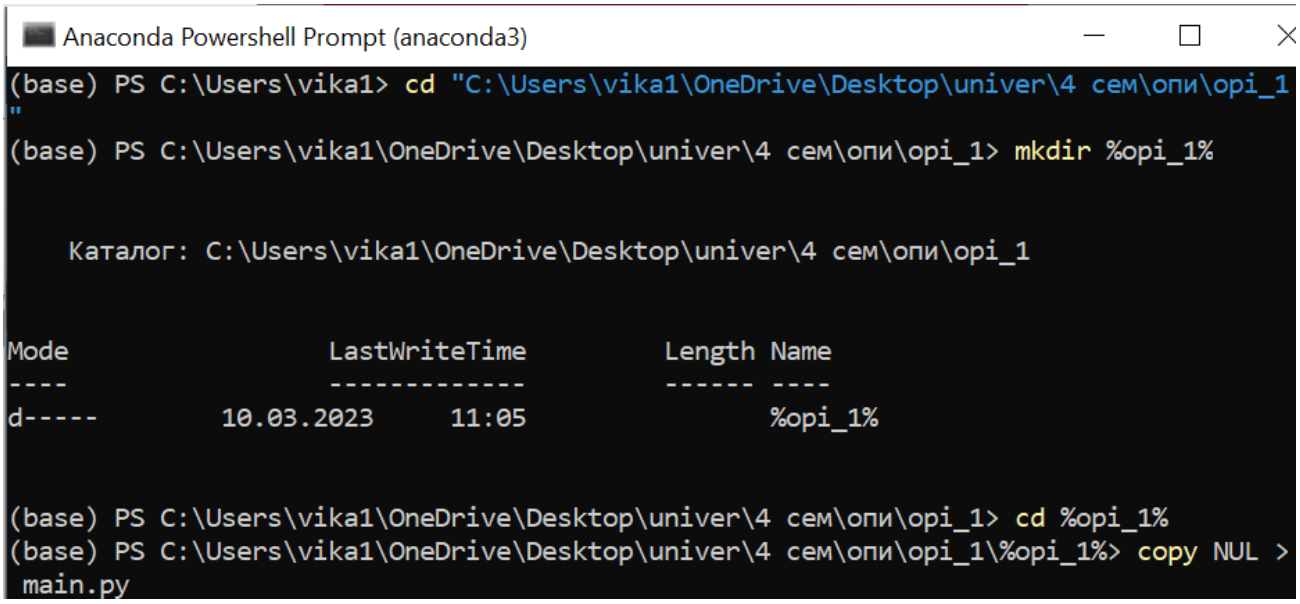

Рисунок 11– Создание чистого вирт. окр. с conda

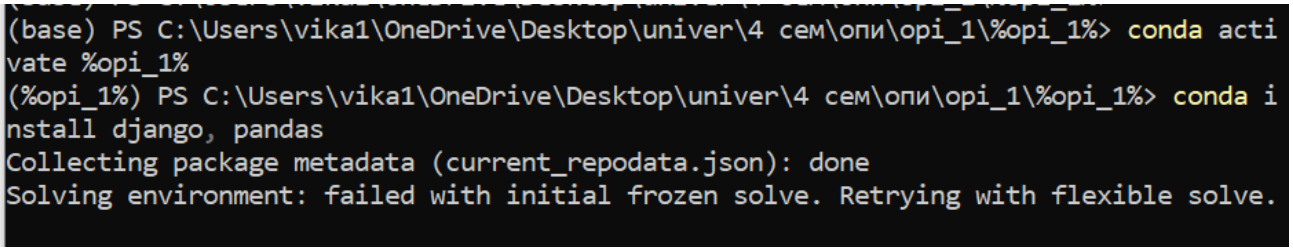

Рисунок 11– Активация виртуального окружения, установка пакетов django и pandas

```
(base) PS C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\%opi_1%> conda crea
te -n %opi_1% python=3.7
Collecting package metadata (current_repodata.json): done
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 22.9.0
  latest version: 23.1.0

Please update conda by running

    $ conda update -n base -c defaults conda



## Package Plan ##

  environment location: C:\Users\vika1\anaconda3\envs\%opi_1%

  added / updated specs:
    - python=3.7


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2023.01.10 |       haa95532_0         121 KB
    certifi-2022.12.7          |   py37haa95532_0         149 KB
    openssl-1.1.1t             |       h2bbff1b_0         5.5 MB
    pip-22.3.1                 |   py37haa95532_0         2.7 MB
```

enviroment.yml - Cool File Viewer

```yaml
name: '%opi_1%'
channels:
  - defaults
dependencies:
  - asgiref=3.5.2=py37haa95532_0
  - blas=1.0=mkl
  - bottleneck=1.3.5=py37h080aedc_0
  - ca-certificates=2023.01.10=haa95532_0
  - certifi=2022.12.7=py37haa95532_0
  - django=3.2.15=py37haa95532_0
  - flit-core=3.6.0=pyhd3eb1b0_0
  - intel-openmp=2021.4.0=haa95532_3556
  - krb5=1.19.4=h5b6d351_0
  - libpq=12.9=hb652d5d_3
  - mkl=2021.4.0=haa95532_640
  - mkl-service=2.4.0=py37h2bbff1b_0
  - mkl_fft=1.3.1=py37h277e83a_0
  - mkl_random=1.2.2=py37hf11a4ad_0
  - numexpr=2.8.4=py37h5b0cc5e_0
  - numpy=1.21.5=py37h7a0a035_3
  - numpy-base=1.21.5=py37hca35cd5_3
  - openssl=1.1.1t=h2bbff1b_0
  - packaging=22.0=py37haa95532_0
  - pandas=1.3.5=py37h6214cd6_0
  - pip=22.3.1=py37haa95532_0
  - psycopg2=2.9.3=py37hcd4344a_0
  - python=3.7.16=h6244533_0
  - python-dateutil=2.8.2=pyhd3eb1b0_0
  - pytz=2022.7=py37haa95532_0
  - setuptools=65.6.3=py37haa95532_0
  - six=1.16.0=pyhd3eb1b0_1
  - sqlite=3.40.1=h2bbff1b_0
  - sqlparse=0.4.3=py37haa95532_0
  - typing_extensions=4.4.0=py37haa95532_0
  - vc=14.2=h21ff451_1
  - vs2015_runtime=14.27.29016=h5e58377_2
  - wheel=0.38.4=py37haa95532_0
  - wincertstore=0.2=py37haa95532_2
  - zlib=1.2.13=h8cc25b3_0
prefix: C:\Users\vika1\anaconda3\envs\%opi_1%
```

```
(%opi_1%) PS C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\%opi_1%> conda i
nstall pip,NumPy, Pandas, SciPy
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
(%opi_1%) PS C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\%opi_1%> conda i
nstall TensorFlow
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
```

```
(%opi_1%) PS C:\Users\vika1\OneDrive\Desktop\univer\4 сем\опи\opi_1\%opi_1%> pip ins
tall TensorFlow
Requirement already satisfied: TensorFlow in c:\users\vika1\anaconda3\envs\%opi_1%\l
ib\site-packages (2.10.0)
Collecting tensorflow-io-gcs-filesystem>=0.23.1
  Downloading tensorflow_io_gcs_filesystem-0.31.0-cp37-cp37m-win_amd64.whl (1.5 MB)
     ---------------------------------------- 1.5/1.5 MB 3.3 MB/s eta 0:00:00
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\vika1\anaconda3\envs\
%opi_1%\lib\site-packages (from TensorFlow) (0.2.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in c:\users\vika1\anaconda
3\envs\%opi_1%\lib\site-packages (from TensorFlow) (1.1.2)
Requirement already satisfied: h5py>=2.9.0 in c:\users\vika1\anaconda3\envs\%opi_1%\
lib\site-packages (from TensorFlow) (3.7.0)
Collecting protobuf<3.20,>=3.9.2
```

ВОПРОСЫ

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

Существует так называемый Python Package Index (PyPI) – это репозито- рий, открытый для всех Python разработчиков, в нем вы можете найти пакетыдля решения практически любых задач.

2. Как осуществить установку менеджера пакетов pip?

При развертывании современной версии Python, pip устанавливается ав-томатически. Но если, по какой-то причине, pip не установлен на вашем ПК, то сделать это можно вручную. Чтобы установить pip, нужно скачать скрипт get-pip.py и выполнить его.

3. Откуда менеджер пакетов pip по умолчанию устанавливает па-кеты?

По умолчанию менеджер пакетов pip скачивает пакеты из Python Package Index (PyPI).

4. Как установить последнюю версию пакета с помощью pip?

С помощью команды $ pip install ProjectName.

5. Как установить заданную версию пакета с помощью pip?

С помощью команды $ pip install ProjectName==3.2, где вместо 3.2 необ-ходимо указать нужную версию пакета.

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

С помощью команды $ pip install e git+https://gitrepo.com/ ProjectName.git

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

С помощью команды $ pip install ./dist/ProjectName.tar.gz

8. Как удалить установленный пакет с помощью pip?

С помощью команды $ pip uninstall ProjectName можно удалить установленный пакет.

9. Как обновить установленный пакет с помощью pip?

С помощью команды $ pip install --upgrade ProjectName можно обновить необходимый пакет.

10. Как отобразить список установленных пакетов с помощью pip?

Командой $ pip list можно отобразить список установленных пакетов.

11. Каковы причины появления виртуальных окружений в языке Python?

Существует несколько причин появления виртуальных окружений в языке Python - проблема обратной совместимости и проблема коллективной разработки. Проблема обратной совмести - некоторые операционные системы, например, Linux и MacOs используют содержащиеся в них предустановлен-ные интерпретаторы Python. Обновив или изменив самостоятельно версию ка-когото установленного глобально пакета, мы можем непреднамеренно сло-мать работу утилит и приложений из дистрибутива операционной системы.

Проблема коллективной разработки - Если разработчик работает над проектом не один, а с командой, ему нужно передавать и получать список зависимостей, а также обновлять их на своем компьютере таким образом, чтобы не нарушалась работа других его проектов. Значит нам нужен механизм, который вместе с обменом проектами быстро устанавливал бы локально и все необходимые для них пакеты, при этом не мешая работе других проектов.

12. Каковы основные этапы работы с виртуальными окружениями?

Основные этапы:

Создаём через утилиту новое виртуальное окружение в отдельной папк для выбранной версии интерпретатора Python.

Активируем ранее созданное виртуального окружения для работы.

Работаем в виртуальном окружении, а именно управляем пакетами ис-пользуя pip и запускаем выполнение кода.

Деактивируем после окончания работы виртуальное окружение.

Удаляем папку с виртуальным окружением, если оно нам больше ненужно.

13. Как осуществляется работа с виртуальными окружениями с по-мощью venv?

С его помощью можно создать виртуальную среду, в которую можно устанавливать пакеты независимо от основной среды или других виртуальных окружений. Основные действия с виртуальными окружениями с помощью venv: создание виртуального окружения, его активация и деактивация.

14. Как осуществляется работа с виртуальными окружениями с по- мощью virtualenv?

Для начала пакет нужно установить. Установку можно выполнить ко- мандой: python3 -m pip install virtualenv Virtualenv позволяет создать абсо- лютно изолированное виртуальное окружение для каждой из программ.

Окружением является обычная директория, которая содержит копию всего необходимого для запуска определенной программы, включая копию са-мого интерпретатора, полной стандартной библиотеки, pip, и, что самое глав-ное, копии всех необходимых пакетов.

15. Изучите работу с виртуальными окружениями pipenv. Как осу-ществляется работа с виртуальными окружениями pipenv?

Для формирования и развертывания пакетных зависимостей использу- ется утилита pip.

Основные возможности pipenv:

– Создание и управление виртуальным окружением

– Синхронизация пакетов в Pipfile при установке и удалении пакетов

– Автоматическая подгрузка переменных окружения из .env файла

После установки pipenv начинается работа с окружением. Его можно со- здать в

любой папке. Достаточно установить любой пакет внутри папки.

Используем requests, он автоматически установит окружение и создаст Pipfile и Pipfile.lock.

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

Установить пакеты можно с помощью команды: pip install –r requirements.txt. Также можно использовать команду pip freeze > requirements.txt, которая создаст requirements.txt наполнив его названиями и версиями тех пакетов что используются вами в текущем окружении. Это удобно если вы разработали проект и в текущем окружении все работает, но вы хотите перенести проект в иное окружением (например, заказчику или на сервер). С помощью закрепления зависимостей мы можем быть уверены, что пакеты, установленные в нашей производственной среде, будут точно соот- ветствовать пакетам в нашей среде разработки, чтобы ваш проект неожиданно не ломался.

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

Conda способна управлять пакетами как для Python, так и для C/ C++, R, Ruby, Lua, Scala и других. Conda устанавливает двоичные файлы, поэтому ра- боту по компиляции пакета самостоятельно выполнять не требуется (по срав-нению с pip).

18. В какие дистрибутивы Python входит пакетный менеджер conda?

Все чаще среди Python-разработчиков заходит речь о менеджере пакетов conda, включенный в состав дистрибутивов Anaconda и Miniconda. JetBrains включил этот инструмент в состав PyCharm.

19. Как создать виртуальное окружение conda?

С помощью команды: conda create -n %PROJ_NAME% python=3.7

20. Как активировать и установить пакеты в виртуальное окруже-ниеconda?

Чтобы установить пакеты, необходимо воспользоваться командой: –conda install A для активации: conda activate %PROJ_NAME%

21. Как деактивировать и удалить виртуальное окружение conda?

Для деактивации использовать команду: conda deactivate, а для удале-ния: conda remove -n $PROJ_NAME.

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

Создание файла: conda env export > environment.yml

Файл environment.yml позволит воссоздать окружение в любой нужныймомент.

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

Достаточно набрать: conda env create -f environment.yml

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

Работа с виртуальными окружениями в PyCharm зависит от способа взаимодействия с виртуальным окружением:

Создаём проект со своим собственным виртуальным окружением, куда затем будут устанавливаться необходимые библиотеки.

Предварительно создаём виртуальное окружение, куда установим нуж- ные библиотеки. И затем при создании проекта в PyCharm можно будет его выбирать, т.е. использовать для нескольких проектов.

Для первого способа ход работы следующий: запускаем PyCharm и в окне приветствия выбираем Create New Project.В мастере создания проекта, указываем в поле Location путь расположения создаваемого проекта. Имя ко- нечной директории также является именем проекта. Далее разворачиваем па раметры окружения, щелкая по Project Interpreter. И выбираем New environ- ment using Virtualenv. Путь расположения окружения генерируется автомати- чески. И нажимаем на Create. Теперь установим библиотеки, которые будем использовать в программе. С помощью главного меню переходим в настройки File → Settings. Где переходим в Project: project_name → Project Interpreter. Вы- ходим из настроек. Для запускапрограммы, необходимо создать профиль с конфигурацией. Для этого в верхнем правом углу нажимаем на кнопку Add Configuration. Откроется окно Run/Debug Configurations, где нажимаем на кнопку с плюсом (Add New Configuration) в правом верхнем углу и выбираем Python. Далее указываем в поле Name имя конфигурации и в поле Script path расположение Python файла с кодом программы. В завершение нажимаем наApply, затем на OK. Для второго способа необходимо сделать следующее: на экране приветствия в нижнем правом углу через Configure → Settings перехо- дим в настройки. Затем переходим в раздел Project Interpreter.

В верхнем правом углу есть кнопка с шестерёнкой, нажимаем на неё и выбираем Add, создавая новое окружение. И указываем расположение для но-вого окружения. Нажимаем на ОК. Далее в созданном окружении устанавли-ваем нужные пакеты. И выходим из настроек. В окне приветствия выбираем Create New Project. В мастере создания проекта, указываем имя расположения проекта в поле Location. Разворачиваем параметры окружения, щелкая по Project Interpreter, где выбираем Existing interpreter и указываем нужное нам окружение. Далее создаем конфигурацию запуска программы, также как со-здавали для раннее. После чего можно выполнить программу.

25. Почему файлы requirements.txt и environment.yml должны хра- ниться в репозитории git?

Чтобы пользователи, которые скачивают какие-либо программы, скрипты, модули могли без проблем посмотреть, какие пакеты им нужно уста-новить дополнительно для корректной работы. За описание о наличии каких- либо пакетов в среде как раз и отвечают файлы requirements.txt и environment.yml.