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 №1
дисциплины «Основы программной инженерии»

Выполнил:
Мелтонян Одиссей
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:
Кандидат технических наук, доцент
кафедры инфокоммуникаций
Воронкин Р. А

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2024 г

Тема: Модули и пакеты

Цель работы: приобретение навыков по работе с модулями и пакетами языка программирования Python

Ход работы:

1. Изучен теоретический материал работы.

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

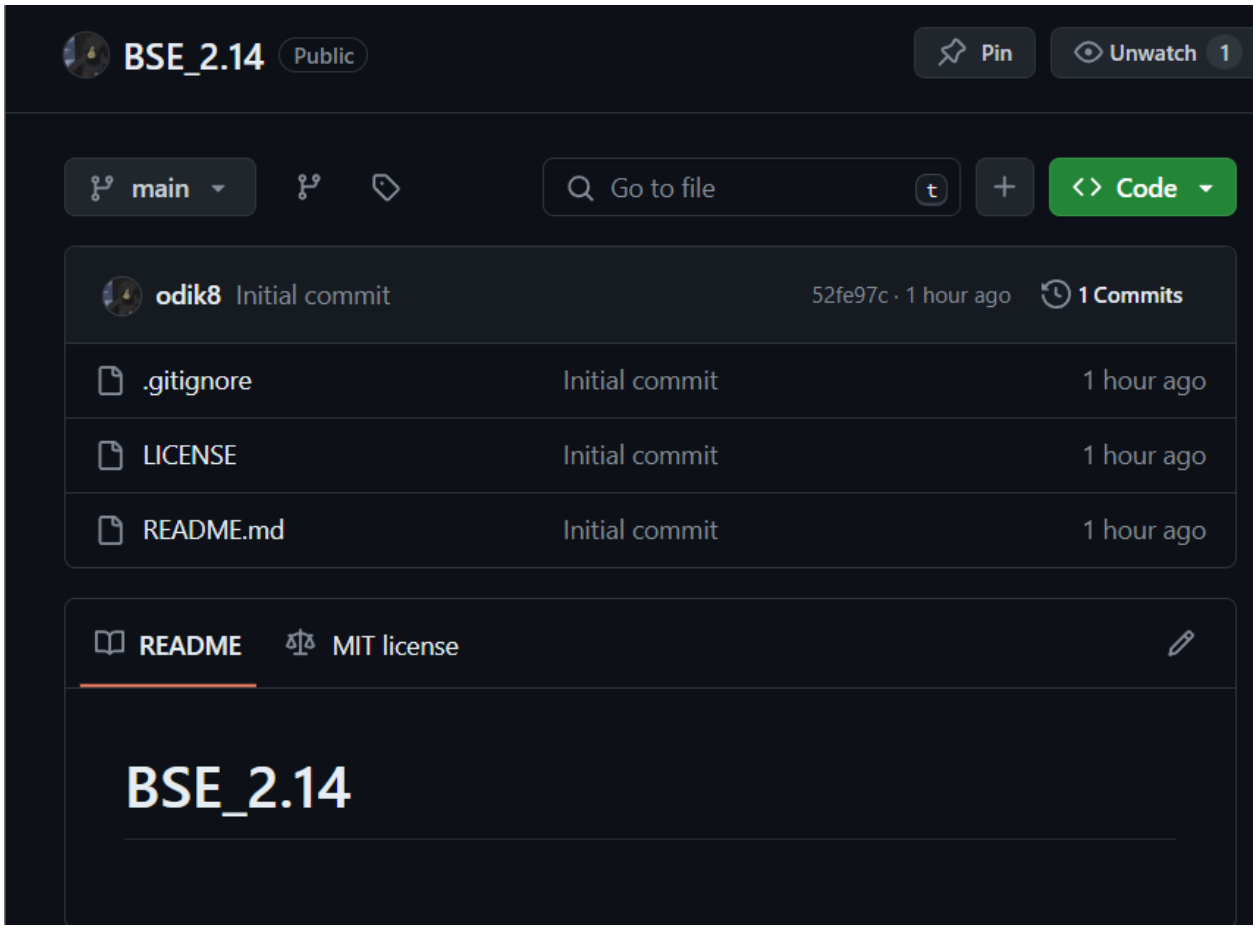

Рисунок 1 – Созданный репозиторий

3. Выполнено клонирование созданного репозитория.

4. Организован репозиторий в соответствие с моделью ветвления git-flow.

```
varfe@DESKTOP-E1O8KEH MINGW64 ~/Рабочий стол/4 семестр/ОПИ/ЛР2(2) (develop)
$ git flow log
Comparing against "master" branch\n

varfe@DESKTOP-E1O8KEH MINGW64 ~/Рабочий стол/4 семестр/ОПИ/ЛР2(2) (develop)
$
```

5. Командой conda create -n bse_2.14 создано ВО

```
(base) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>conda info --envs
# conda environments:
#
base                  *  C:\Users\varfe\anaconda3
bse_2.14                 C:\Users\varfe\anaconda3\envs\bse_2.14
```

Рисунок 2 – Список ВО

6. Активировано созданное ВО Anaconda

```
(base) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>conda activate bse_2.14

(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>
```

Рисунок 3 – Активация ВО

7. Установлены в виртуальное окружение следующие пакеты: pip, NumPy, Pandas, SciPy.

```
(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>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\varfe\anaconda3\envs\bse_2.14

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


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    bottleneck-1.3.7           |  py312he558020_0         131 KB
    expat-2.6.2                |       hd77b12b_0         260 KB
    mkl-service-2.4.0          |  py312h2bbff1b_1          55 KB
    mkl_fft-1.3.8              |  py312h2bbff1b_0         160 KB
    mkl_random-1.2.4           |  py312h59b6b97_0         196 KB
    numexpr-2.8.7              |  py312h96b7d27_0         144 KB
    numpy-1.26.4               |  py312hfd52020_0          11 KB
    numpy-base-1.26.4          |  py312h4dde369_0         6.6 MB
    pandas-2.2.1               |  py312h0158946_0        14.2 MB
    pip-23.3.1                 |  py312haa95532_0         2.9 MB
    python-3.12.3              |       h1d929f7_0        16.4 MB
    pytz-2023.3.post1          |  py312haa95532_0         199 KB
    scipy-1.12.0               |  py312hbb039d4_0        22.9 MB
    setuptools-68.2.2          |  py312haa95532_0         1.2 MB
    wheel-0.41.2               |  py312haa95532_0         150 KB
    ------------------------------------------------------------
                                           Total:        65.5 MB
```

Рисунок 4 – Установка пакетов

8. Установлен пакет tensorflow

```
(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>conda install tensorflow
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
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 3.12.* , 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 5 – Попытка установки пакета

Для установки tensorflow требуется python версии 3.11 и ниже

Для этого было создано новое ВО с нужной версией python

```
(base) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>conda create -n bse_2.14 python=3.11
```

```
tensorflow                2.10.0          mkl_py39ha510bab_0
tensorflow-base           2.10.0          mkl_py39h6a7f48e_0
tensorflow-estimator      2.10.0           py39haa95532_0
termcolor                 2.1.0            py39haa95532_0
tk                        8.6.12               h2bbff1b_0
typing_extensions         4.9.0            py39haa95532_1
tzdata                    2024a                h04d1e81_0
urllib3                   2.1.0            py39haa95532_1
vc                        14.2                 h21ff451_1
vs2015_runtime            14.27.29016          h5e58377_2
werkzeug                  2.3.8            py39haa95532_0
wheel                     0.41.2           py39haa95532_0
win_inet_pton             1.1.0            py39haa95532_0
wrapt                     1.14.1           py39h2bbff1b_0
xz                        5.4.6                h8cc25b3_0
yarl                      1.9.3            py39h2bbff1b_0
zipp                      3.17.0           py39haa95532_0
zlib                      1.2.13               h8cc25b3_0

(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>
```

Рисунок 6 – Список установленных пакетов

6. Установлен TensorFlow с помощью менеджера пакетов pip

```
(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>pip install tensorflow
```

Рисунок 7 – Установка TensorFlow с помощью менеджера пакетов pip

7. Сформированы файлы environment.yml. и requirements.txt.

```
(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>conda env export > environment.yml
(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>pip freeze > requirements.txt
(bse_2.14) C:\Users\varfe\Рабочий стол\4 семестр\ОПИ\ЛР2(2)>
```

Рисунок 8 – Формирование файлов

Файл environment.yml содержит информацию о имени окружения, используемых каналах и зависимостях (пакеты и их версии), необходимых для создания точно такого же окружения.

```
name: bse_2.14
channels:
  - defaults
dependencies: ...
prefix: C:\Users\varfe\anaconda3\envs\bse_2.14
```

Рисунок 9 – Файл environment.yml

Файл requirements.txt содержит список всех зависимостей проекта и их версий.

8. Зафиксированы сделанные изменения. Выполнено слияние ветки develop с master и оправлены на GitHub

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? – Скачать скрипт и выполнить его:

$ curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py

3. Откуда менеджер пакетов pip по умолчанию устанавливает пакеты? – Существует так называемый Python Package Index (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач. Там также есть возможность выкладывать свои пакеты.

4. Как установить последнюю версию пакета с помощью pip? – Последнюю версию пакета можно установить, не указывая версии:

pip install имя_пакета

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

pip install имя_пакета==версия

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

pip install git+URL_репозитория

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

pip install /путь/к/пакету

8. Как удалить установленный пакет с помощью pip? –

pip uninstall имя_пакета

9. Как обновить установленный пакет с помощью pip? –

pip install --upgrade имя_пакета

10. Как отобразить список установленных пакетов с помощью pip?

pip list

11. Каковы причины появления виртуальных окружений в языке Python? – Виртуальные окружения используются для изоляции зависимостей и проектных настроек между различными проектами, чтобы избежать конфликтов и обеспечить чистоту среды разработки.

12. Каковы основные этапы работы с виртуальными окружениями? – Создание, активация, установка зависимостей, деактивация и удаление.

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 Linux и macOS нам нужно дать команду:

source env/bin/activate

Чтобы активировать виртуальное окружение под Windows команда выглядит иначе:

> env\\Scripts\\activate

Просто под Windows мы вызываем скрипт активации напрямую.

Чтобы деактивировать:

deactivate

source /home/user/envs/project1_env2/bin/activate

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

Установку можно выполнить командой:

Для python 3

python3 -m pip install virtualenv

Для единственного python

python -m pip install virtualenv

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

Создание в текущей папке виртуального окружения для интерпретатора доступного через команду python3 с названием папки окружения env:

virtualenv -p python3 env

Активация и деактивация такая же, как у стандартной утилиты Python. Например, для Linux и macOS:

$ source env/bin/activate

(env) $ deactivate

Для операционной системы Windows:

> env\\Scripts\\activate

(env) > deactivate

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

Файл requirements.txt содержит в себе в список зависимостей

Создание:

pip freeze > requirements.txt

Формат: .txt

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?

conda входит в дистрибутивы Anaconda и Miniconda

19. Как создать виртуальное окружение conda?

conda create -n %PROJ_NAME% python=<version>

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

Активация:

conda activate %PROJ_NAME%

Установка пакетов:

conda install <имя_пакета>, <имя_пакета>*…

21. Как деактивировать и удалить виртуальное окружение conda?

Деактивация:

conda deactivate

Удаление:

conda remove -n $PROJ_NAME

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

Файл environment.yml используется для описания окружения и его зависимостей в conda.

Создание:

conda env export > environment.yml.

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

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.

25. Почему файлы requirements.txt и environment.yml должны храниться в репозитории git? – Файлы requirements.txt и environment.yml должны храниться в репозитории Git для обеспечения воспроизводимости среды разработки и упрощения процесса установки зависимостей для других разработчиков.