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 РАБОТЕ №2.14
дисциплины «Основы кроссплатформенного программирования»

Выполнил:
Сластёнов Андрей Сергеевич
1 курс, группа ИТС-б-о-21-1,
11.03.02 «Инфокоммуникационные технологии и системы связи»,
направленность (профиль) Сети связи и системы коммутации,
очная форма обучения

_____
(подпись)

Руководитель практики от университета:
Воронкин Р.А., канд. тех. наук, доцент,
доцент кафедры инфокоммуникаций

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2023 г

Тема: уловные операторы и циклы в языке Python.

Цель работы: приобретение навыков программирования разветвляющихся алгоритмов и алгоритмов циклической структуры. Освоить операторы языка Python версии 3.x if, while, for, break и continue, позволяющих реализовывать разветвляющиеся алгоритмы и алгоритмы циклической структуры.

Порядок выполнения работы:

1. Создал новый репозиторий и клонировал его на свой компьютер.

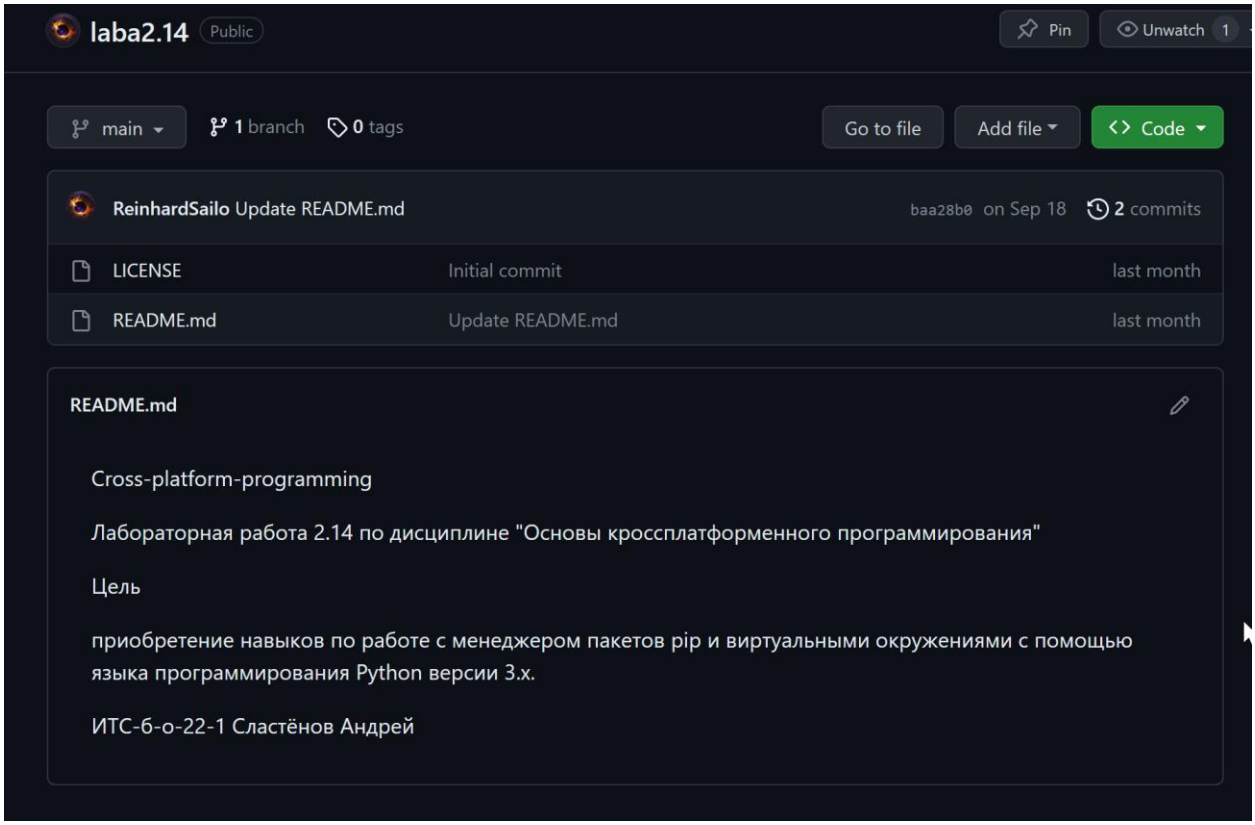

Рис 1. Новый репозиторий

2. Клонировал репозиторий на свой компьютер. В ходе данной лабораторной работы работала с моделью ветвления git-flow.

```
PS E:\Programming\laba2.14> 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]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [E:/Programming/laba2.14/.git/hooks]
PS E:\Programming\laba2.14> git status
On branch develop
nothing to commit, working tree clean
PS E:\Programming\laba2.14>
```

Рис 2. Клонирование и модель ветвления git-flow

3. Создал виртуальное окружение Anaconda с именем репозитория.

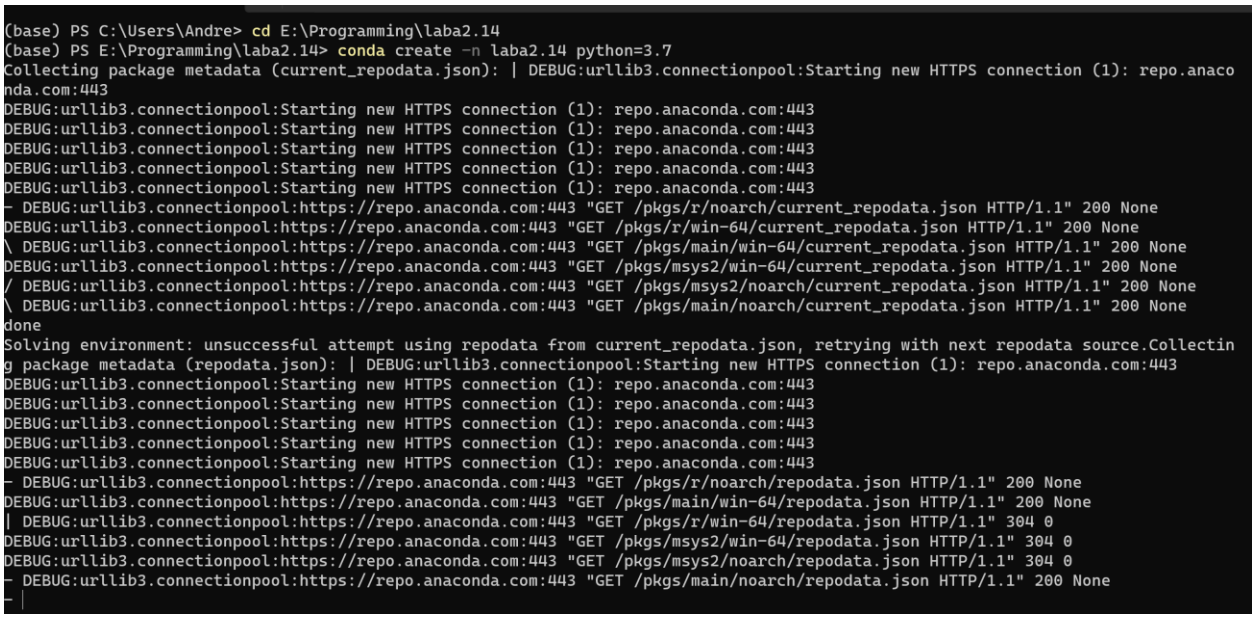

Рис 3. Создание виртуального окружения

4. Установил в виртуальное окружение следующие пакеты: pip, NumPy, Pandas, SciPy.

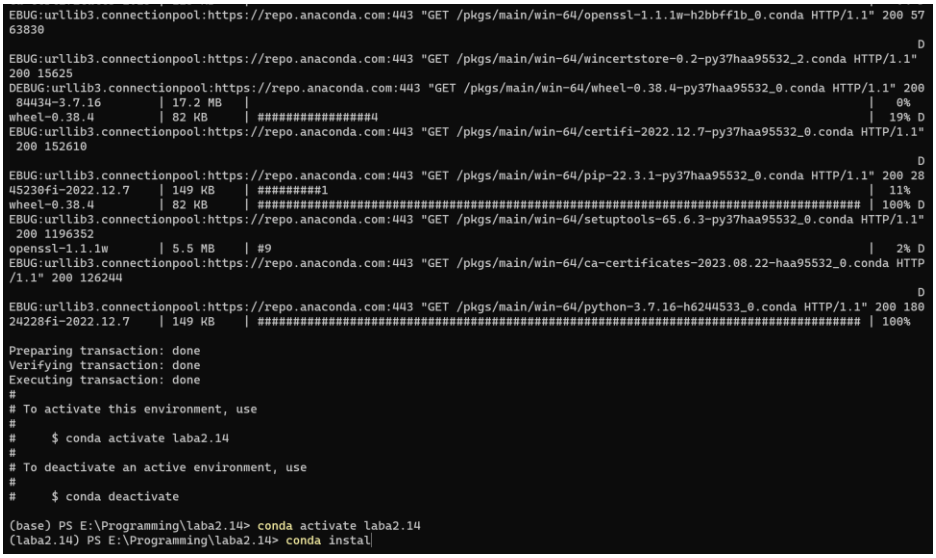

Рис 4. Установка пакетов в виртуальное окружение

5. Сформировал файлы requirements.txt и environment.yml.

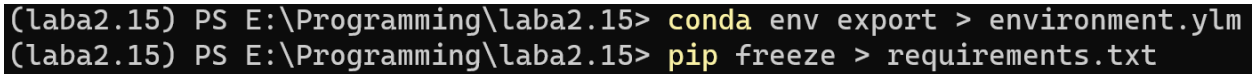

Рис 5. Файлы requirements.txt и environment.yml

6. Зафиксировал изменения.

```
PS E:\Programming\laba2.15> git add .
PS E:\Programming\laba2.15> git commit -m "add files"
[main d3e15dc] add files
 8 files changed, 75 insertions(+)
 create mode 100644 Ex.1.py
 create mode 100644 Ex.2.py
 create mode 100644 Ex.3.py
 create mode 100644 Individual.1.py
 create mode 100644 Individual.2.py
 create mode 100644 environment.ylm
 create mode 100644 file2.txt
 create mode 100644 requirements.txt
PS E:\Programming\laba2.15> git checkout main
Already on 'main'
```

Рис 6. Изменения в ветке develop

7. Слил ветку develop с веткой main и отправил на удаленный сервер.

```
PS E:\Programming\laba2.15> git checkout main
Already on 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
PS E:\Programming\laba2.15> git merge develop
merge: develop - not something we can merge
PS E:\Programming\laba2.15> git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 3.31 KiB | 3.31 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ReinhardSailo/laba2.15.git
   5ef6e8a..d3e15dc  main -> main
PS E:\Programming\laba2.15>
```

Рис 8. Слияние веток develop в main и push

Ссылка: https://github.com/ReinhardSailo/laba2.14

Ответы на контрольные вопросы

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

Существует так называемый PythonPackageIndex (PyPI) – это репозиторий, открытый для всех Python разработчиков, в нем вы можете найти пакеты для решения практически любых задач.

2. Как осуществить установку менеджера пакетов pip? python -m pip< аргументы>

3. Откуда менеджер пакетов pip по умолчанию устанавливает пакеты?

4. Как установить последнюю версию пакета с помощью pip?python -m pip install -U pip

5. Как установить заданную версию пакета с помощью pip? pipinstallProjectName==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? pipuninstallProjectName

9. Как обновить установленный пакет с помощью pip? pipinstall --upgradeProjectName

10. Как отобразить список установленных пакетов с помощью pip?piplist

11. Каковы причины появления виртуальных окружений в языке Python?

В отдельной папке создаётся неполная копия выбранной установки Python. Эта копия является просто набором файлов (например, интерпретатора или ссылки на него), утилит для работы с собой и несколькихпакетов (в том числе pip). Стандартные пакеты при этом не копируются.

12. Каковы основные этапы работы с виртуальными окружениями?

1) Создаём через утилиту новое виртуальное окружение в отдельной папке для выбранной

2) версии интерпретатора Python.

3) Активируем ранее созданное виртуального окружения для работы.

4) Работаем в виртуальном окружении, а именно управляем пакетами используя pip и

5) запускаем выполнение кода.

6) Деактивируем после окончания работы виртуальное окружение.

7) Удаляем папку с виртуальным окружением, если оно нам больше не нужно.

13. Как осуществляется работа с виртуальными окружениями с

помощью venv?

Создав виртуальное окружение в папке проекта. После её выполнения создастся папка env с виртуальным окружением. После активации приглашение консоли изменится. В его начале в круглых скобках будет отображаться имя папки с виртуальным окружением.

14. Как осуществляется работа с виртуальными окружениями спомощью virtualenv?

Создание в текущей папке виртуального окружения для интерпретатора доступного через команду python3 с названием папки окружения env. reeze - команда, используемая для получения всех установленных пакетов в формате требований. Таким образом, все пакеты, которые вы установили перед выполнением команды и предположительно использовалив каком-либо проекте, будут перечислены в файле с именем «requirements.txt». Кроме того, будут указаны их точные версии

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

При запуске проект с Pipenv, он автоматически создает виртуальную среду для текущего проекта, даже если вы еще не используете ее. Pipenv управляет зависимостями, отказавшись от привычного requirements.txt, и заменяя его на новый документ под названием Pipfile.

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

Способ хранения списка внешних зависимостей проекта. pipfreeze>; requirements.txt . < requirementspecifier>

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

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? Anaconda

19. Как создать виртуальное окружение conda? conda create -n %PROJ_NAME% python=3.7 condaactivate %PROJ_NAME%

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

condaactivateenv

21. Как деактивировать и удалить виртуальное окружение conda?

condadeactivate

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

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

Создайте окружение из environment.yml файла Первая строка yml файла задает имя новой среды. Активируйте новую среду: conda activate my env Убедитесь, что новая среда установлена правильно: condaenvlist

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

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

Стандартные файлы виртуального окружения.

Вывод: приобрел навыки по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.