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 №2.14**

**дисциплины**

**«Основы кроссплатформенного программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Пинаева Диана Олеговна  2 курс, группа ИТС-б-о-22-1,  11.03.02 «Инфокоммуникационные технологии и системы связи», направленность (профиль) «Инфокоммуникационные системы и сети», очная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  Воронкин Р. А., доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема:** Установка пакетов в Python. Виртуальные окружения.

**Цель:** приобретение навыков по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x.

# Ход работы:

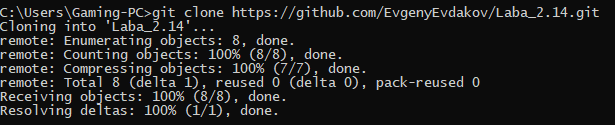
Задание 1**.** Создала общедоступный репозиторий на GitHub, в котором использована лицензий MIT и язык программирования Python, также добавила файл .gitignore с необходимыми правилами. Клонировала свой репозиторий на свой компьютер.

Рисунок 1. Клонирование репозитория

Задание 2.Организовала свой репозиторий в соответствие с моделью ветвления git-flow, появилась новая ветка develop в которой буду выполнять дальнейшие задачи.

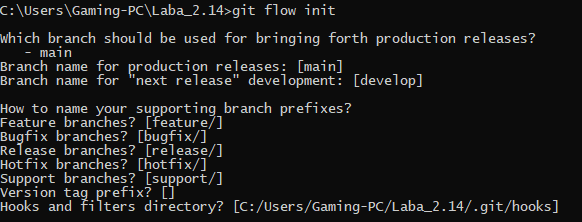


Рисунок 2. Модель ветвления git-flow

Задание 3.Создала виртуальное окружение Anaconda с именем репозитория.

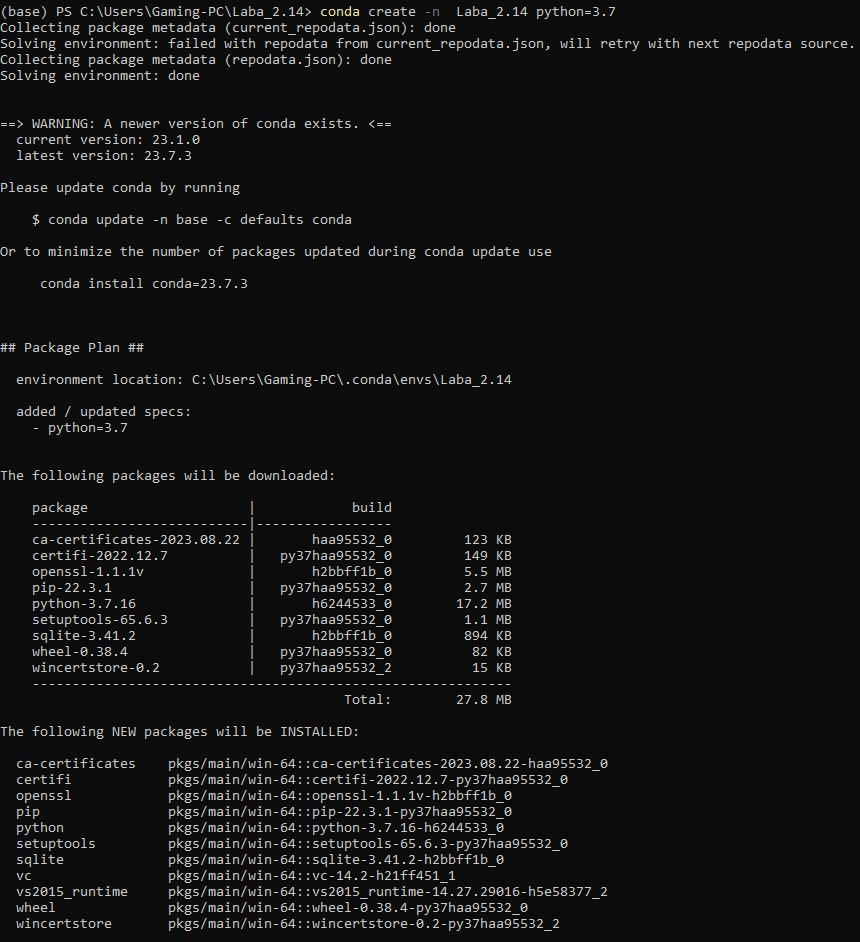


Рисунок 3. Создание виртуального окуружения

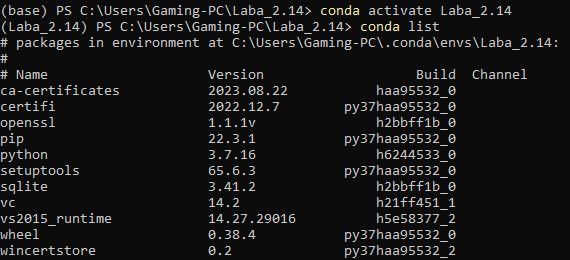


Рисунок 4. Активация окружения

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

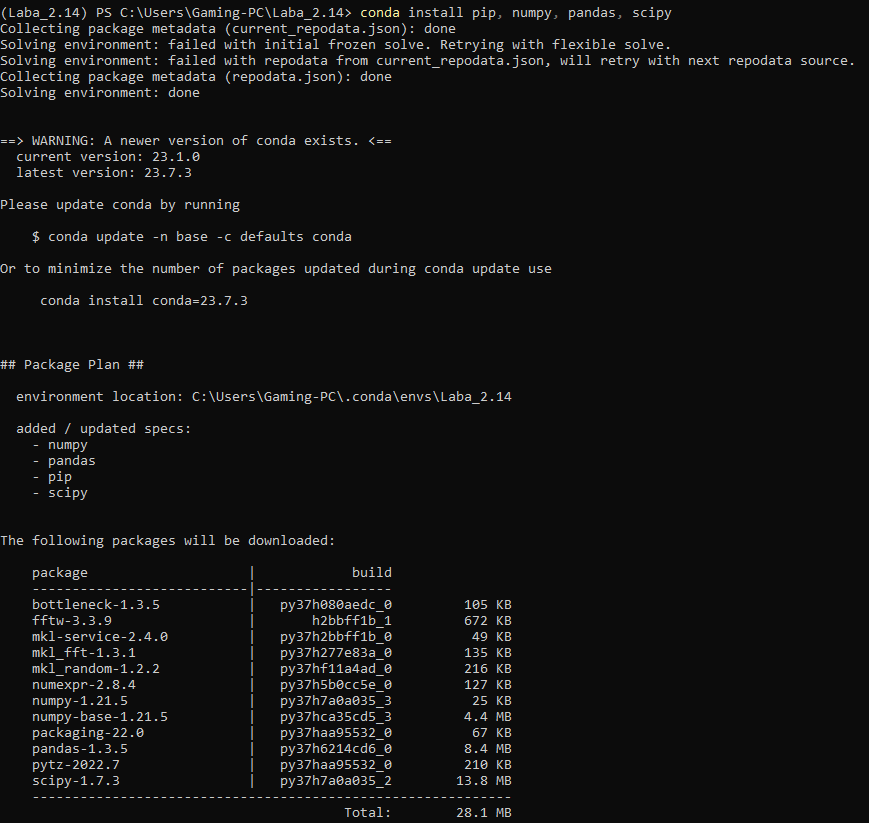


Рисунок 5. Установка пакетов

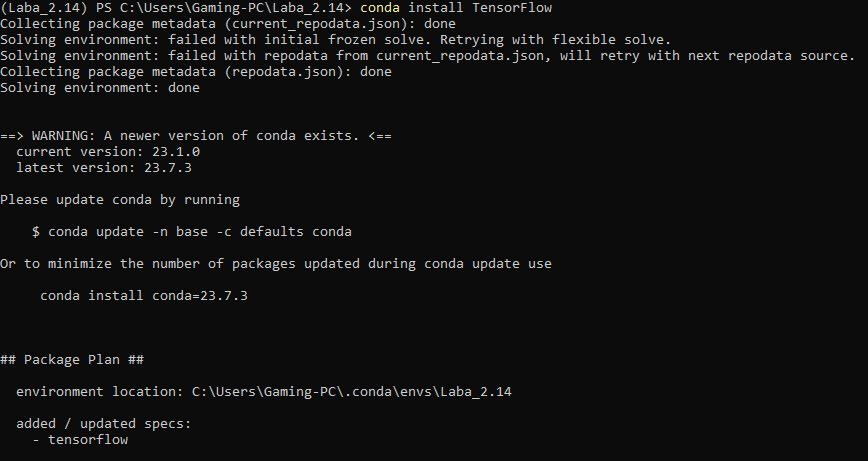
Задание 5.Попыталась установить менеджером пакетов conda пакет TensorFlow. В результате выдало предупреждение о разных версиях.

Рисунок 6. Установка TensorFlow через conda

После попытки установки пакета TensorFlow через conda, воспользовалась менеджером пакетов pip. После чего выполнила установку данного пакета.



Рисунок 7. Установка TensorFlow через pip

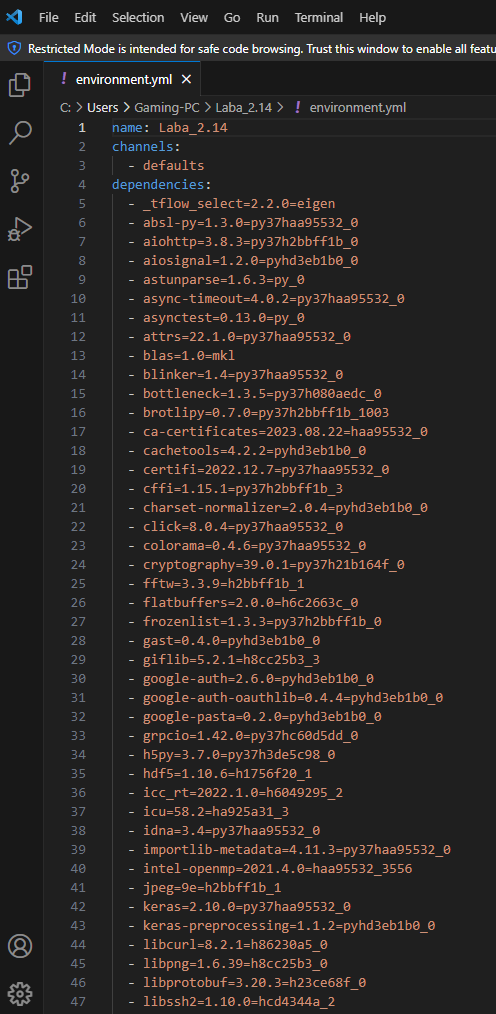
Задание 6.Сформировала файл environment.yml, он определяет окружение conda, включая название, каналы (источники пакетов) и список зависимостей. Он позволяет указать не только Python-пакеты, но и другие зависимости (например, библиотеки, необходимые для компиляции), а также версию Python.

Рисунок 8. Содержимое файла

Далее сформировала файл requirements.txt он содержит список зависимостей и их версий в текстовом формате. Каждая строка представляет собой один пакет с указанием версии или диапазона версий.

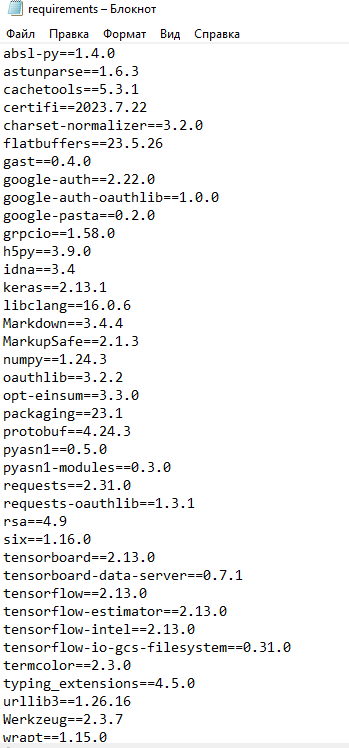
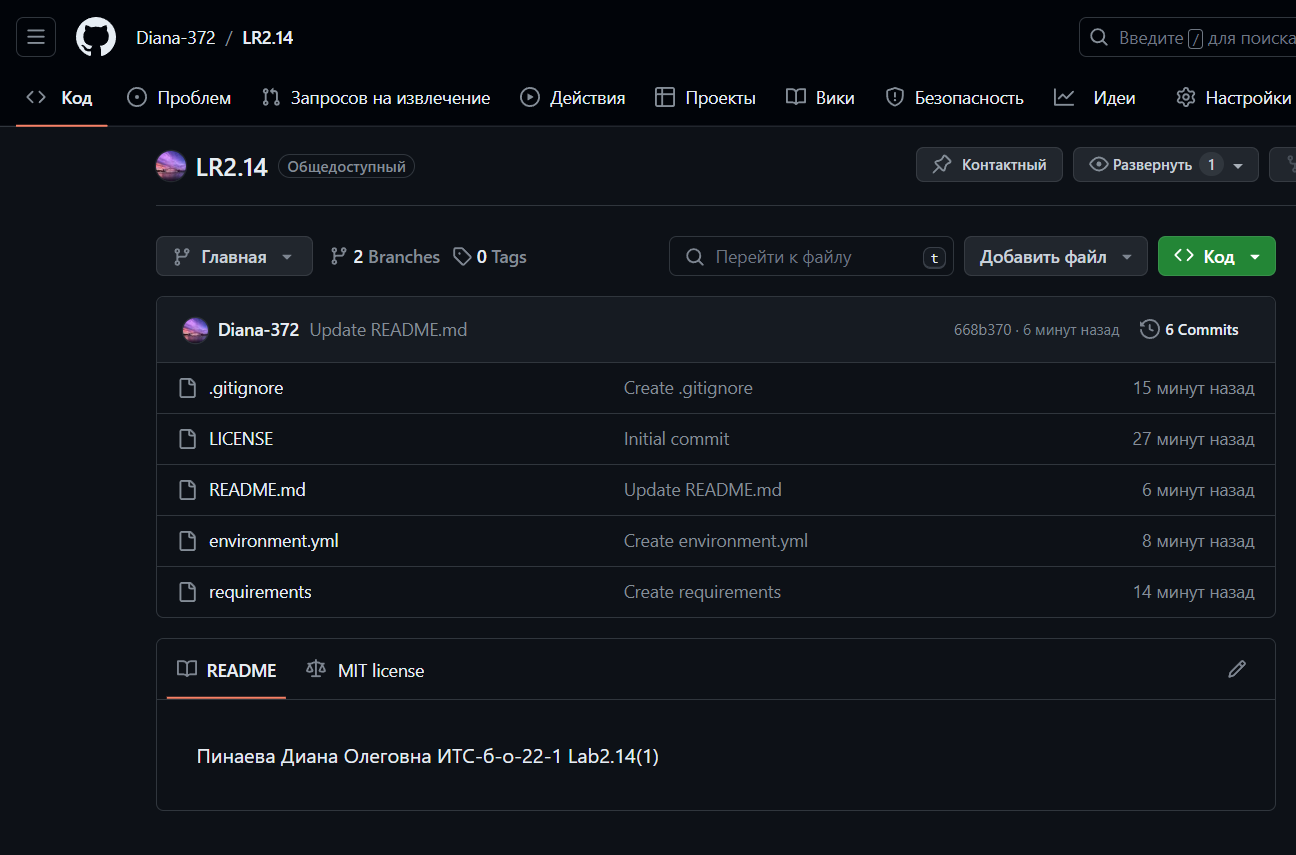


Рисунок 9. Содержимое файла requirements.txt

Задание 7.Закомитила изменения и отправила на GitHub.



Ссылка: <https://github.com/Diana-372/LR2.14.git>

# Ответы на контрольные вопросы:

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

Для установки пакета Python, не входящего в стандартную библиотеку, можно воспользоваться менеджером пакетов pip. Для этого нужно выполнить команду pip install package\_name

# Как осуществить установку менеджера пакетов pip?

Pip обычно устанавливается автоматически вместе с Python. Если он не установлен можно использовать команду python -m ensurepip --default-pip

# Откуда менеджер пакетов pip по умолчанию устанавливает пакеты?

По умолчанию, pip устанавливает пакеты из Python Package Index (PyPI)

# Как установить последнюю версию пакета с помощью pip?

Для установки последней версии пакета с помощью pip, нужно выполнить команду pip install --upgrade package\_name

# Как установить заданную версию пакета с помощью pip?

Для установки заданной версии пакета с помощью pip, нужно выполнить команду pip install package\_name==desired\_version

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

Для установки пакета из git репозитория, можно использовать команду pip install git+URL

# Как установить пакет из локальной директории с помощью pip?

Для установки пакета из локальной директории, нужно выполнить

команду pip install /path/to/package\_directory

# Как удалить установленный пакет с помощью pip?

Чтобы удалить установленный пакет с помощью pip, нужно выполнить команду pip uninstall package\_name

# Как обновить установленный пакет с помощью pip?

Для обновления установленного пакета с помощью pip, нужно использовать команду pip install --upgrade package\_name

# Как отобразить список установленных пакетов с помощью pip?

Чтобы отобразить список установленных пакетов с помощью pip, нужно выполнить команду pip list или pip freeze.

# Каковы причины появления виртуальных окружений в языке Python?

Виртуальные окружения в Python используются для изоляции проектов и их зависимостей. Основные причины их использования включают изоляцию проектов друг от друга, управление версиями зависимостей и предотвращение конфликтов между пакетами.

# Каковы основные этапы работы с виртуальными окружениями?

1. Создание виртуального окружения.
2. Активация виртуального окружения.
3. Установка необходимых пакетов в виртуальное окружение.
4. Работа с проектом в активированном виртуальном окружении.
5. Деактивация виртуального окружения (по завершении работы).

# Как осуществляется работа с виртуальными окружениями с помощью venv?

1. Создание виртуального окружения: python -m venv myenv
2. Активация виртуального окружения
3. Установка пакетов: pip install package\_name
4. Деактивация: deactivate

# Как осуществляется работа с виртуальными окружениями с помощью virtualenv?

Работа с виртуальными окружениями с помощью virtualenv аналогична venv. Создаем окружение с помощью virtualenv myenv, активируем его, устанавливаем пакеты и деактивируем.

# Изучите работу с виртуальными окружениями pipenv. Как осущестляется работа с виртуальными окружениями pipenv?

1. Создание виртуального окружения: pipenv --python 3.8 (здесь указывается версия Python)
2. Установка пакетов: pipenv install package\_name
3. Запуск оболочки в виртуальном окружении: pipenv shell

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

Файл requirements.txt используется для указания списка зависимостей проекта. Создать его можно с помощью команды pip freeze > requirements.txt, и он будет содержать список установленных пакетов и их версий.

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

Преимущества conda по сравнению с pip включают в себя возможность установки бинарных зависимостей, управление средами и зависимостями, а также поддержку многих языков программирования, не только Python.

# В какие дистрибутивы Python входит пакетный менеджер conda?

Пакетный менеджер conda входит в дистрибутивы Anaconda и Miniconda. Он также может быть установлен отдельно на другие дистрибутивы Python.

# Как создать виртуальное окружение conda?

Для создания виртуального окружения conda нужно использовать команду conda create --name myenv

# Как активировать и установить пакеты в виртуальное окружение conda?

Для активации и установки пакетов в виртуальное окружение conda, нужно выполнить команду conda activate myenv, а затем команду conda install package\_name для установки пакетов.

# Как деактивировать и удалить виртуальное окружение conda?

Чтобы деактивировать и удалить виртуальное окружение conda, нужно выполнить команду conda deactivate для деактивации, а затем conda env remove --name myenv для удаления.

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

Файл environment.yml используется для определения окружения conda, включая зависимости. Его можно создать вручную, указав пакеты и их версии, или автоматически с помощью команды conda env export > environment.yml

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

Для создания виртуального окружения conda с помощью файла environment.yml, нужно использовать команду conda env create -f environment.yml

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

В среде PyCharm можно создавать и управлять виртуальными окружениями conda через интерфейс IDE. Это делается через "File" -> "Settings" -> "Project: <your project>" -> "Python Interpreter

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

Файлы requirements.txt и environment.yml содержат информацию о зависимостях проекта и их версиях. Хранение их в репозитории Git позволяет другим разработчикам воссоздать окружение проекта и установить необходимые зависимости для работы с кодом.

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