1. Используя команду cat в терминале операционной системы Linux, создать
два файла
Домашние животные (заполнив файл собаками, кошками, хомяками)
Ответ:
```
$ cat <<EOT >> Домашние_животные.txt
> собака
> кошка
> хомяк
> кролик
> EOT
```

и Вьючные животными заполнив файл Лошадьми, верблюдами и ослы),
Ответ:
```
$ cat <<EOT >> Вьючные_животные.txt
> лошадь
> верблюд
> осел
> EOT
```

а затем объединить их.
Ответ:
```
$ cat  Домашние_животные.txt Вьючные_животные.txt >> объединенный_файл.txt
```

Просмотреть содержимое созданного файла.
Ответ:
```
$ cat Домашние_животные.txt
```

Переименовать файл, дав ему новое имя (Друзья человека).
Ответ:
```
$ mv объединенный_файл.txt Друзья_человека.txt
```

2. Создать директорию,
Ответ:
```
$ mkdir new_dir
```

переместить файл туда
Ответ:
```
$ mv Друзья_человека.txt ./new_dir
```

3. Подключить дополнительный репозиторий MySQL. Установить любой пакет
из этого репозитория
Ответ:
```
$ sudo apt install mysql-server
```

4. Установить
Ответ:
```
$ sudo dpkg - i /home/rusttm/Downloads/mysql-apt-config_0.8.26-1_all.deb
```

и удалить deb-пакет с помощью dpkg.
Ответ:
$ sudo dpkg -P mysql

5. Выложить историю команд в терминале ubuntu

6. Нарисовать диаграмму, в которой есть класс родительский класс, домашние
животные и вьючные животные, в составы которых в случае домашних животных войдут классы: собаки, кошки, хомяки, а в класс вьючные животные
войдут: Лошади, верблюды и ослы).
(см. Animals_classes_diagram.pdf)

7. В подключенном MySQL репозитории создать базу данных “Друзья человека”
Ответ:
```
CREATE DATABASE IF NOT EXISTS Друзья_человека;
```

8. Создать таблицы с иерархией из диаграммы в БД
Ответ:
```
CREATE TABLE IF NOT EXISTS animals
        (animal_group VARCHAR(255) NOT NULL PRIMARY KEY);
CREATE TABLE IF NOT EXISTS pets
        (animal_type VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_group VARCHAR(255) DEFAULT 'pets', FOREIGN KEY
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS packs
        (animal_type VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_group VARCHAR(255) DEFAULT 'packs', FOREIGN KEY
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS dogs
        (animal_name VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_commands TEXT,
        animal_birth Date,
        animal_type VARCHAR(255) DEFAULT 'dogs', FOREIGN KEY
(animal_type) REFERENCES pets(animal_type),
        animal_group VARCHAR(255) DEFAULT 'pets',  FOREIGN KEY
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS cats
        (animal_name VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_commands TEXT,
        animal_birth Date,
        animal_type VARCHAR(255) DEFAULT 'cats', FOREIGN KEY
(animal_type) REFERENCES pets(animal_type),
        animal_group VARCHAR(255) DEFAULT 'pets',  FOREIGN KEY
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS hamsters
        (animal_name VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_commands TEXT,
        animal_birth Date,
        animal_type VARCHAR(255) DEFAULT 'hamsters', FOREIGN KEY
(animal_type) REFERENCES pets(animal_type),
        animal_group VARCHAR(255) DEFAULT 'pets',  FOREIGN KEY
```

```
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS horses
        (animal_name VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_commands TEXT,
        animal_birth Date,
        animal_type VARCHAR(255) DEFAULT 'horses', FOREIGN KEY
(animal_type) REFERENCES packs(animal_type),
         animal_group VARCHAR(255) DEFAULT 'packs',  FOREIGN KEY
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS camels
        (animal_name VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_commands TEXT,
        animal_birth Date,
        animal_type VARCHAR(255) DEFAULT 'camels', FOREIGN KEY
(animal_type) REFERENCES packs(animal_type),
        animal_group VARCHAR(255) DEFAULT 'packs',  FOREIGN KEY
(animal_group) REFERENCES animals(animal_group));
CREATE TABLE IF NOT EXISTS donkeys
        (animal_name VARCHAR(255) NOT NULL PRIMARY KEY,
        animal_commands TEXT,
        animal_birth Date,
        animal_type VARCHAR(255) DEFAULT 'donkeys', FOREIGN KEY
(animal_type) REFERENCES packs(animal_type),
        animal_group VARCHAR(255) DEFAULT 'packs',  FOREIGN KEY
(animal_group) REFERENCES animals(animal_group))
```

9. Заполнить низкоуровневые таблицы именами(животных), командами которые они выполняют и датами рождения

```
INSERT INTO animals (animal_group)
VALUES
        ('pets'),
        ('packs');
INSERT INTO pets (animal_type)
VALUES
        ('dogs'),
        ('cats'),
        ('hamsters');
INSERT INTO packs (animal_type)
VALUES
        ('horses'),
        ('camels'),
        ('donkeys');
INSERT INTO dogs (animal_name, animal_commands, animal_birth)
VALUES
        ('Bark', 'sit', '2020-01-31'),
        ('Alf', 'sit', '2022-05-13');
INSERT INTO cats (animal_name, animal_commands, animal_birth)
VALUES
        ('Kittie', 'come', '2023-01-18'),
        ('Leo', 'come eat, ', '2022-02-02');
INSERT INTO hamsters (animal_name, animal_commands, animal_birth)
```

```
VALUES
        ('Fluffy', '', '2023-07-04'),
        ('Lola', '', '2023-08-02');
INSERT INTO horses (animal_name, animal_commands, animal_birth)
VALUES
        ('Amigo', 'come', '2019-01-04'),
        ('Ivy', 'go', '2021-11-27');
INSERT INTO camels (animal_name, animal_commands, animal_birth)
VALUES
        ('Alladin', 'stand', '2018-01-09'),
        ('Sandy', 'stand', '2020-06-24');
INSERT INTO donkeys (animal_name, animal_commands, animal_birth)
VALUES
        ('Oscar', 'stand, go', '2019-06-11'),
        ('Pixie', 'eat', '2023-03-25')
```

10. Удалив из таблицы верблюдов, т.к. верблюдов решили перевезти в другой
питомник на зимовку.
Ответ:

```
DROP TABLE IF EXISTS camels
```

Объединить таблицы лошади, и ослы в одну таблицу.
Ответ:

```
INSERT INTO horses SELECT  * FROM    donkeys d
ON DUPLICATE KEY
UPDATE horses.animal_birth = d.animal_birth
```

11.Создать новую таблицу “молодые животные” в которую попадут все животные старше 1 года, но младше 3 лет
Ответ:

```
CREATE TABLE молодые_животные
AS
SELECT * FROM
(SELECT * FROM cats
UNION SELECT * FROM dogs
UNION SELECT * FROM hamsters
UNION SELECT * FROM horses
UNION SELECT * FROM donkeys)
as new_table
WHERE (animal_birth BETWEEN DATE_SUB( CURDATE( ) ,INTERVAL 3 YEAR )
AND DATE_SUB( CURDATE( ) ,INTERVAL 1 YEAR ))
```

и в отдельном столбце с точностью
до месяца подсчитать возраст животных в новой таблице

```
SELECT animal_birth, TIMESTAMPDIFF(MONTH, animal_birth, CURDATE( ))
AS animal_age
FROM молодые_животные
```

12. Объединить все таблицы в одну, при этом сохраняя поля, указывающие на
прошлую принадлежность к старым таблицам.

Ответ:

```
CREATE TABLE union_table
AS
SELECT * FROM
(SELECT t1.*, null AS animal_age, 'cats' AS base_table_name
FROM cats as t1
UNION ALL
SELECT t2.*, null AS animal_age, 'dogs' AS base_table_name
        FROM dogs as t2
UNION ALL
SELECT t3.*, null AS animal_age, 'hamsters' AS base_table_name
        FROM hamsters as t3
UNION ALL
SELECT t4.*, null AS animal_age, 'horses' AS base_table_name
        FROM horses as t4
UNION ALL
SELECT t5.*, null AS animal_age, 'donkeys' AS base_table_name
        FROM donkeys as t5
UNION ALL
SELECT t6.*, 'молодые_животные' AS base_table_name
    FROM молодые_животные as t6
UNION ALL
SELECT null, null, null, null, t7.*, null AS animal_age, 'animals'
AS base_table_name
    FROM animals as t7
UNION ALL
SELECT null, null, null, t8.*, null AS animal_age, 'pets' AS
base_table_name
    FROM pets as t8
UNION ALL
SELECT null, null, null, t9.*, null AS animal_age, 'packs' AS
base_table_name
    FROM packs as t9
 ) as_temp_table
```

Примечание к пп13-15: Нигде не написано об языке программирования, и я задавал вопрос насчет возможности использования Python.
Ответа так и не получил, поэтому написал на нем.

13.Создать класс с Инкапсуляцией методов и наследованием по диаграмме.
Ответ: class AnimalsMainClass - корневой

14. Написать программу, имитирующую работу реестра домашних животных.
В программе должен быть реализован следующий функционал:

Ответ: написал две программы, обе работают.
А.в папке AnimalsProgram -простая программа для работы с базой данных, работает со словарями. Запуск программы
$ python ./AnimalsProgram/MainClass.py
в папке AnimalsProgram_v2 -сохранений в базу нет, но работа идет со

списком объектов разных классов. Запуск программы
$ python ./AnimalsProgram_v2/MainClass.py

14.1 Завести новое животное
14.2 определять животное в правильный класс
Ответ: не совсем понятно, что имелось ввиду, т.к. классы вроде жестко прописаны
и при заведении нового класс берется автоматически

14.3 увидеть список команд, которое выполняет животное
14.4 обучить животное новым командам
Ответ: сделал в одном меню со списком животных и текущих команд, т.к. для обоих пунктов(14.3 и 14.4) он требуется

14.5 Реализовать навигацию по меню
Ответ: навигация сделана с помощью класса MenuMainClass

15.Создайте класс Счетчик, у которого есть метод add(), увеличивающий
значение внутреннейint переменнойна 1 при нажатие “Завести новое животное” Сделайте так, чтобы с объектом такого типа можно было работать в
блоке try-with-resources. Нужно бросить исключение, если работа с объектом
типа счетчик была не в ресурсном try и/или ресурс остался открыт. Значение
считать в ресурсе try, если при заведения животного заполнены все поля.

Ответ: не совсем ясно, что имелось ввиду?
по-моему, для работы try нужно выбрасывать ошибку,
а зачем ее выбрасывать при незаполненных полях,
когда можно пользователя попросить исправить?

Сделал в виде:
```
try:
    добавить новые команды животному
except:
    пропустить добавление
else:
    увеличить счетчик
```