Итоговая аттестация

Информация о проекте

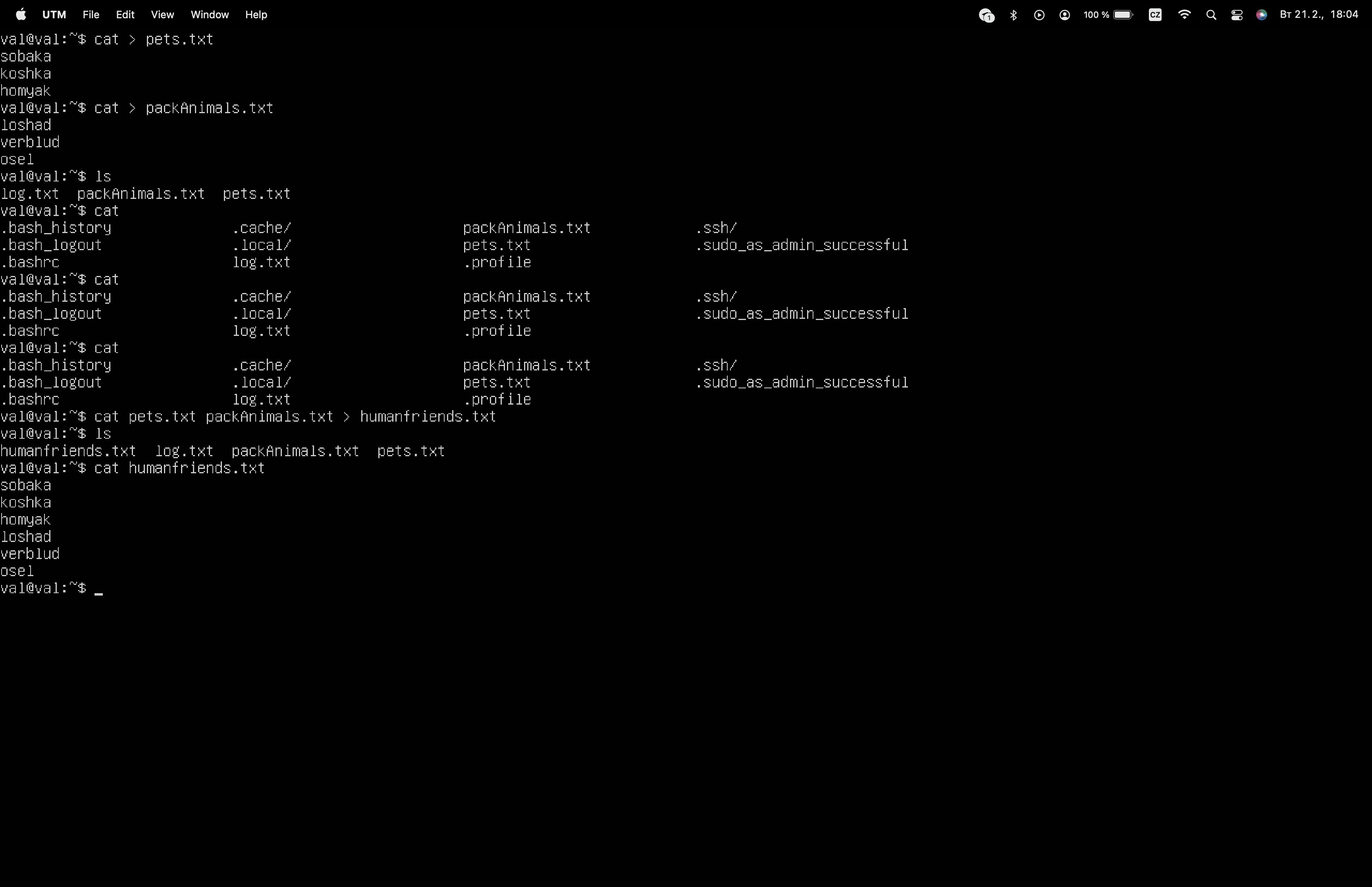
Необходимо организовать систему учета для питомника, в котором живут домашние и вьючные животные.

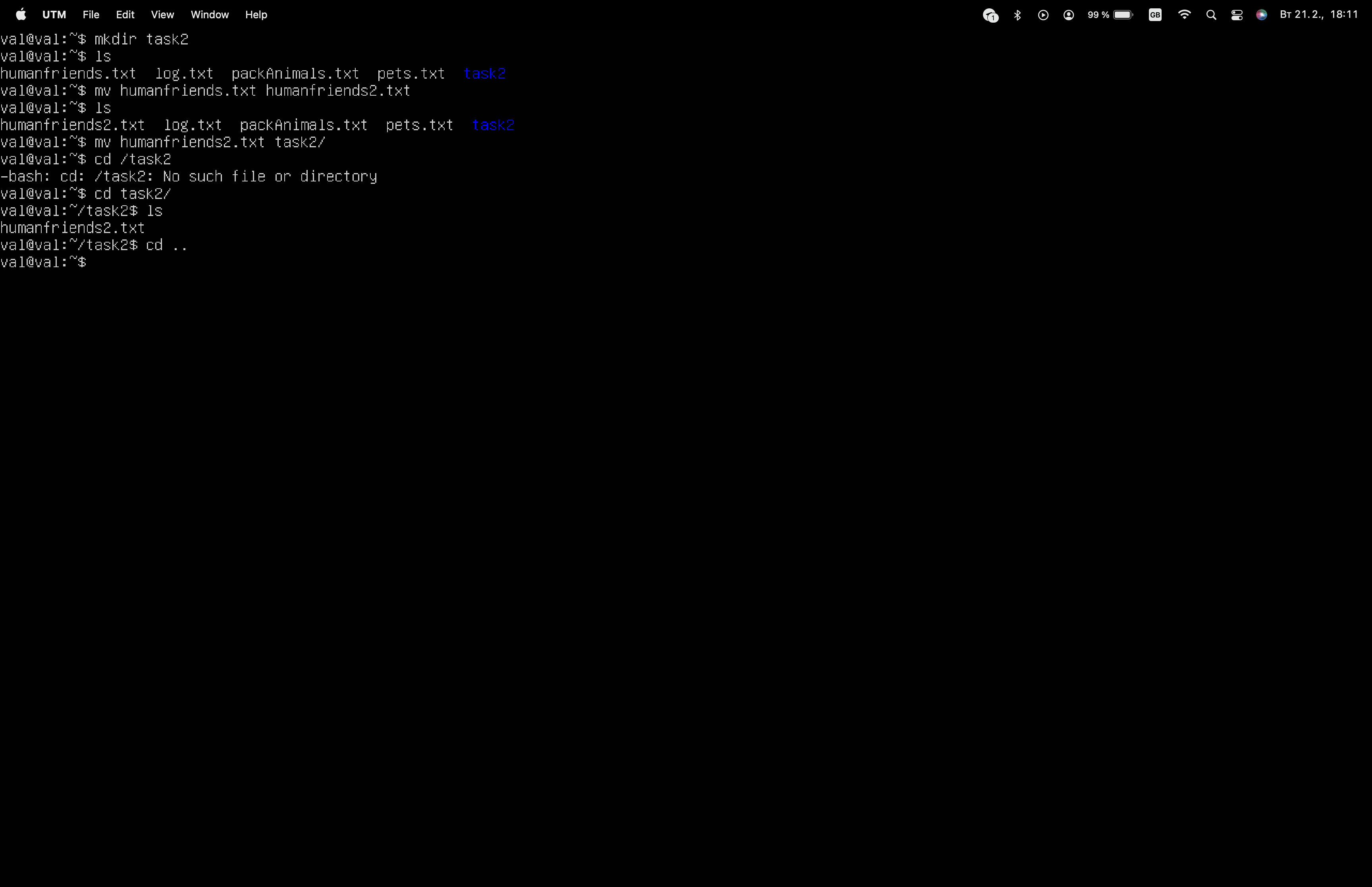
Как сдавать проект

Для сдачи проекта необходимо создать отдельный общедоступный репозиторий(Github, gitlub, или Bitbucket). Разработку вести в этом репозитории, использовать пул реквесты на изменения. Программа должна запускаться и работать, ошибок при выполнении программы быть не должно. Программа, может использоваться в различных системах, поэтому необходимо разработать класс в виде конструктора

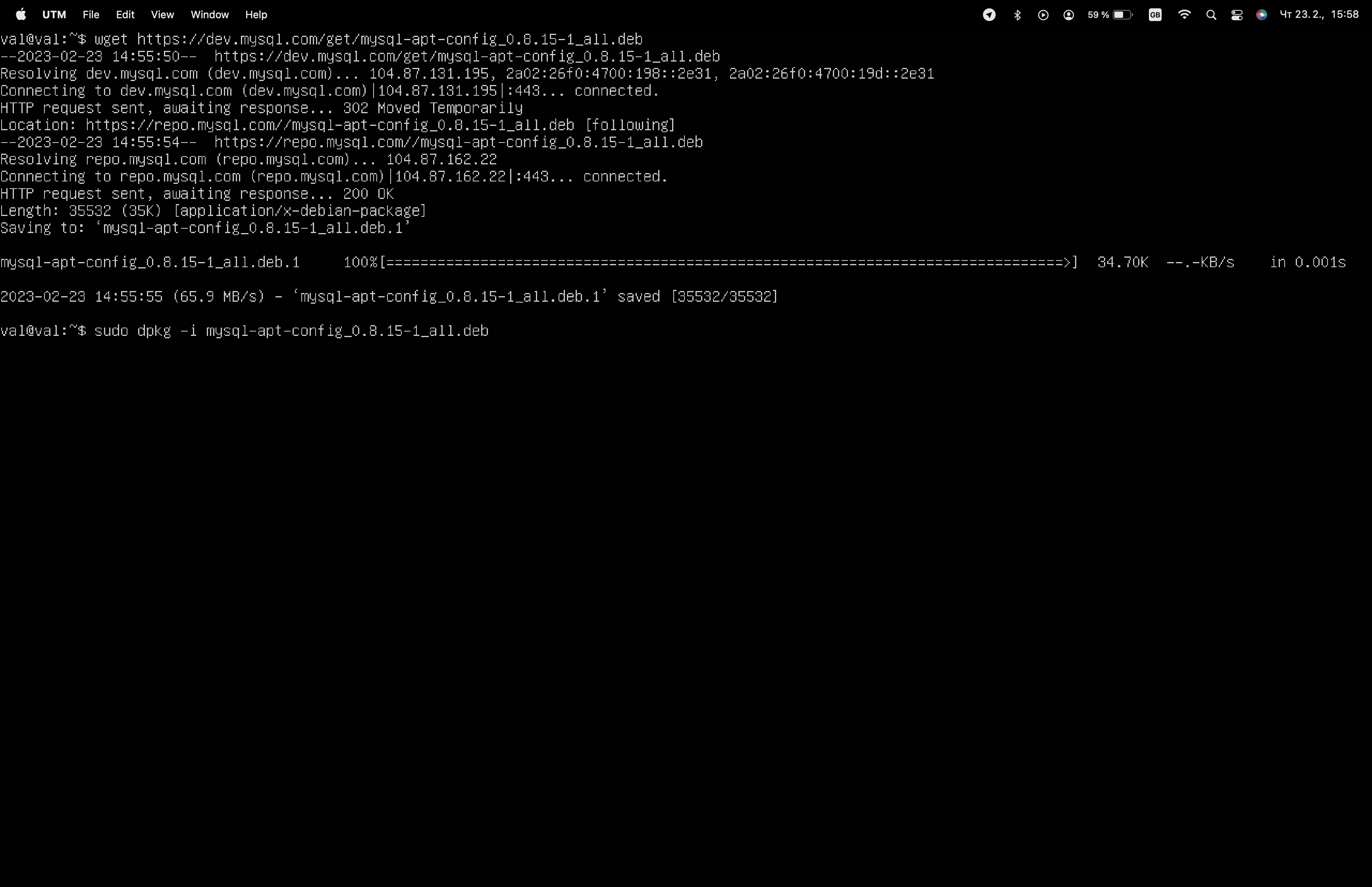
Задание

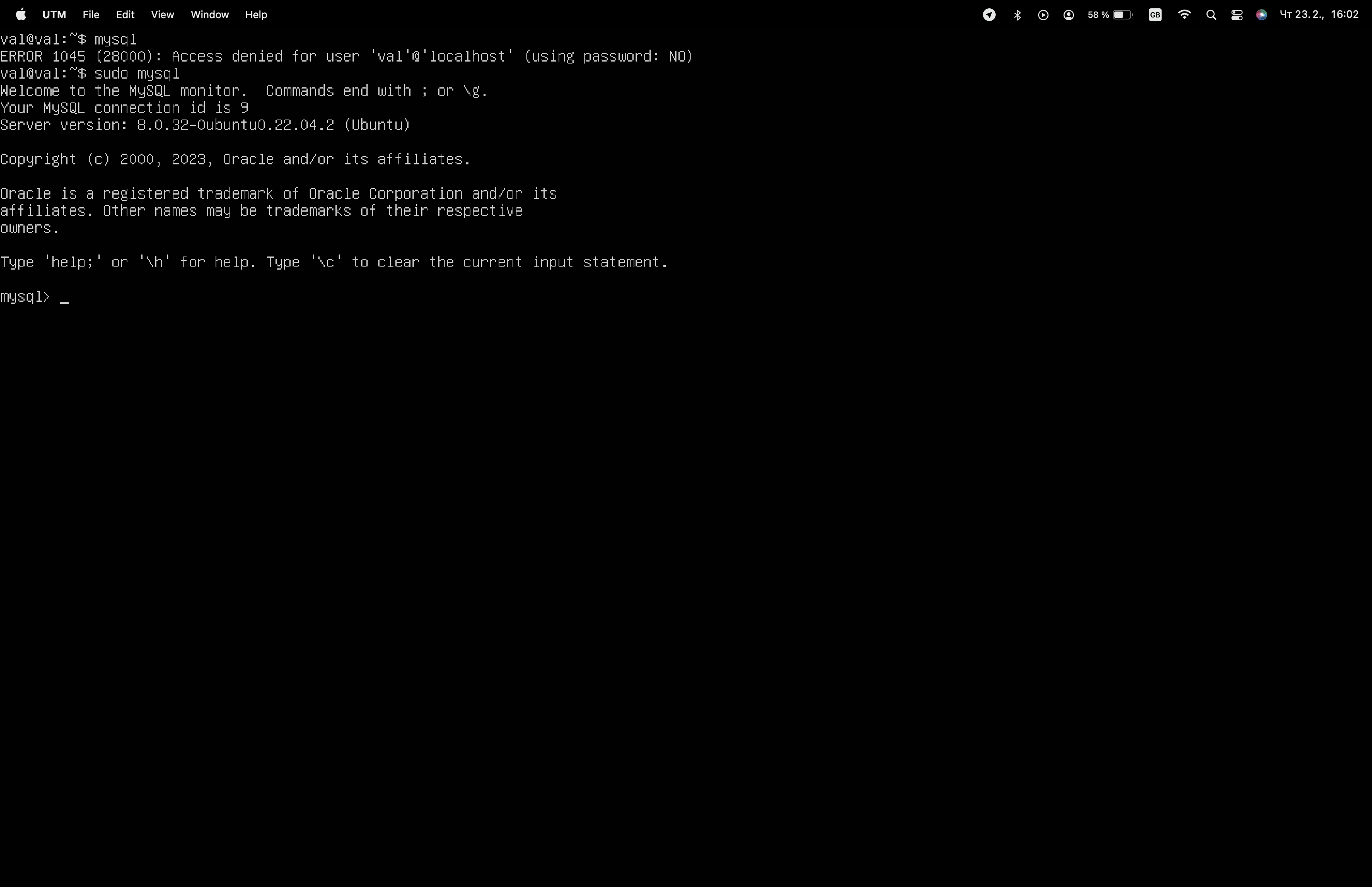
1. Используя команду cat в терминале операционной системы Linux, создать два файла Домашние животные (заполнив файл собаками, кошками, хомяками) и Вьючные животными заполнив файл Лошадьми, верблюдами и ослы), а затем объединить их. Просмотреть содержимое созданного файла. Переименовать файл, дав ему новое имя (Друзья человека).

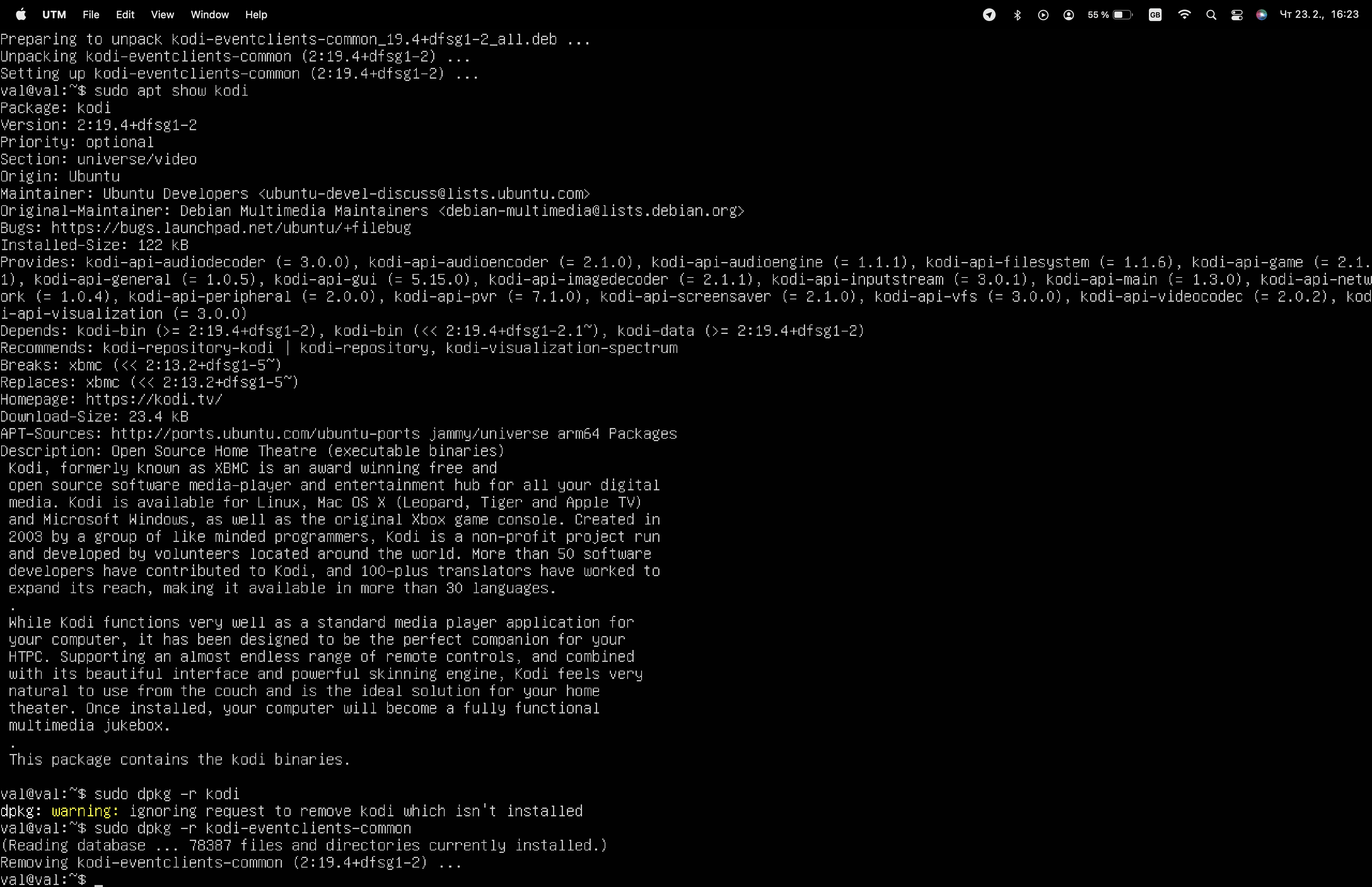
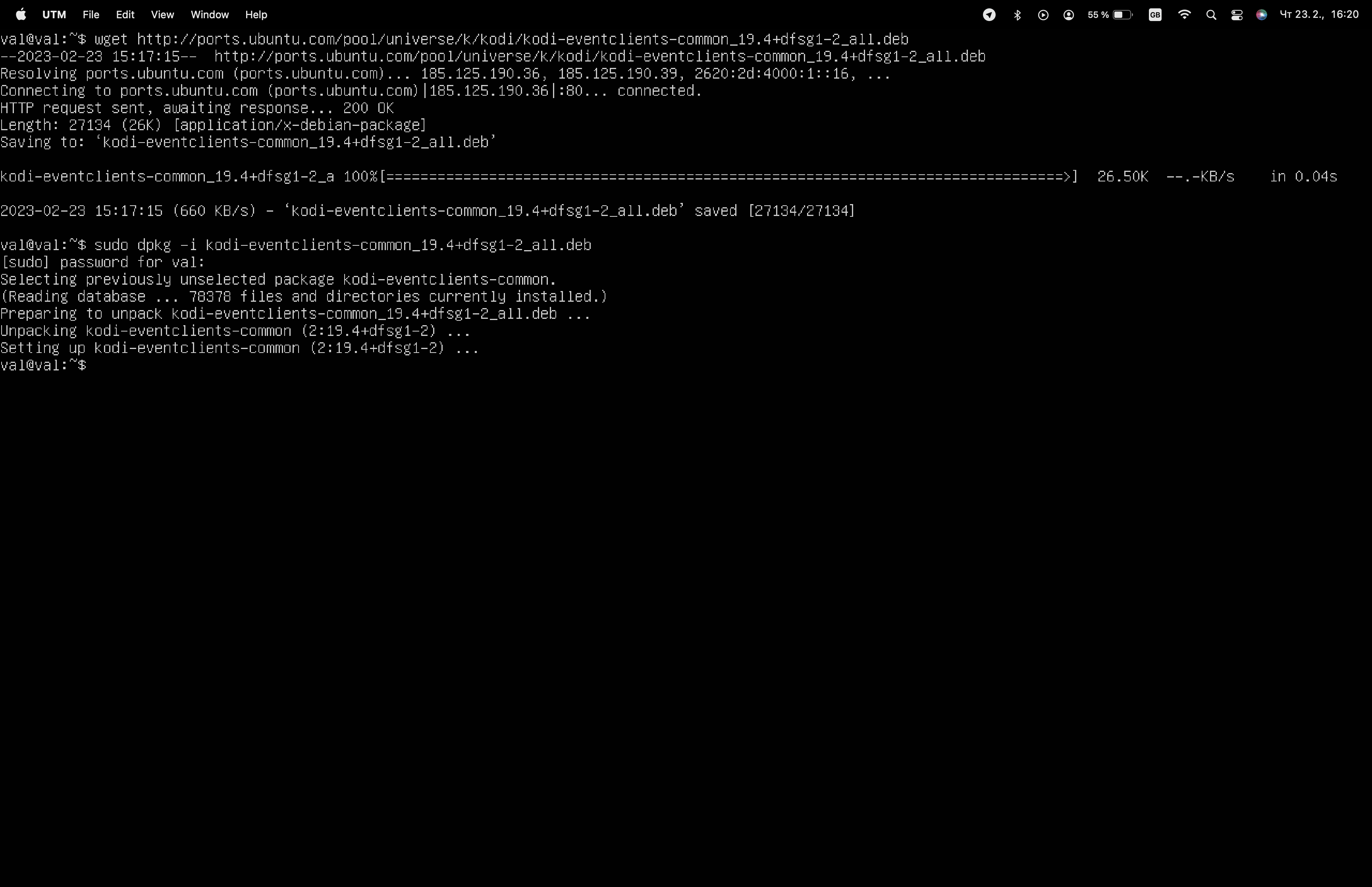


2. Создать директорию, переместить файл туда.

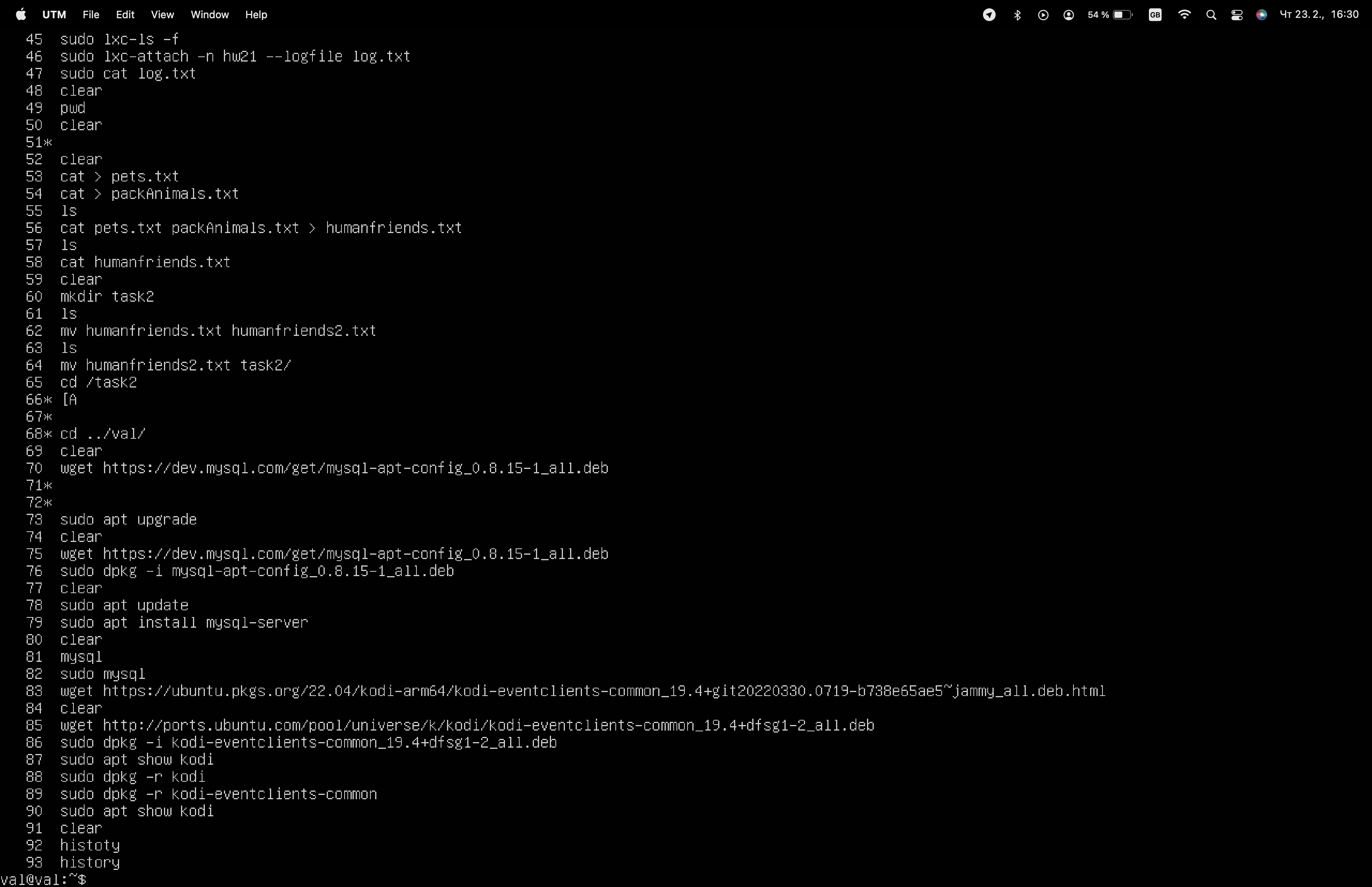
3. Подключить дополнительный репозиторий MySQL. Установить любой пакет из этого репозитория.

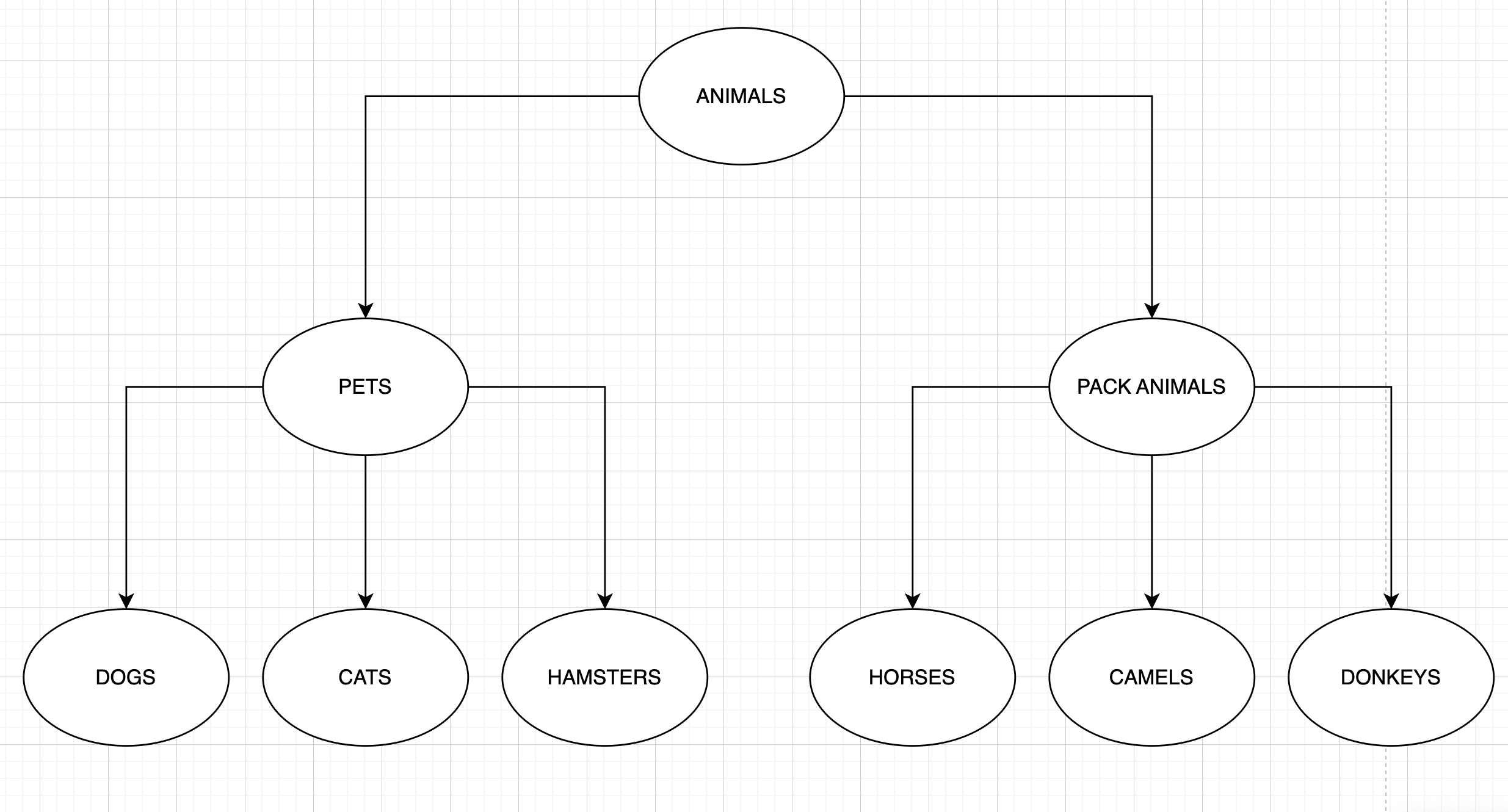




4. Установить и удалить deb-пакет с помощью dpkg.

5. Выложить историю команд в терминале ubuntu

смотреть команды начиная с 53.

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).

7. В подключенном MySQL репозитории создать базу данных “Друзья человека”

8. Создать таблицы с иерархией из диаграммы в БД

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

10. Удалив из таблицы верблюдов, т.к. верблюдов решили перевезти в другой питомник на зимовку. Объединить таблицы лошади, и ослы в одну таблицу.

11. Создать новую таблицу “молодые животные” в которую попадут все животные старше 1 года, но младше 3 лет и в отдельном столбце с точностью до месяца подсчитать возраст животных в новой таблице

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

-- task 7

CREATE DATABASE humanFriends;

USE HumanFriends;

-- task 8

CREATE TABLE Animals

(

ID int auto\_increment primary key,

AnimalsType varchar(30)

);

INSERT INTO Animals(AnimalsType)

VALUES ('Pets'), ('PackAnimals');

CREATE TABLE Pets

(

ID int auto\_increment primary key,

AnimalsTypeID int not null,

PetsType varchar(30),

FOREIGN KEY (AnimalsTypeID) REFERENCES Animals(Id)

ON DELETE CASCADE

);

INSERT INTO Pets(AnimalsTypeId, PetsType)

VALUES (

(SELECT ID FROM Animals WHERE AnimalsType='Pets'),

'Dogs'

),

(

(SELECT ID FROM Animals WHERE AnimalsType='Pets'),

'Cats'

),

(

(SELECT ID FROM Animals WHERE AnimalsType='Pets'),

'Hamsters'

);

SELECT \* FROM PETS;

CREATE TABLE PackAnimals

(

ID int auto\_increment primary key,

AnimalsTypeID int not null,

PackAnimalsType varchar(30),

FOREIGN KEY (AnimalsTypeID) REFERENCES Animals(Id) ON DELETE CASCADE

);

INSERT INTO PackAnimals(AnimalsTypeId, PackAnimalsType)

VALUES (

(SELECT ID FROM Animals WHERE AnimalsType='PackAnimals'),

'Horses'

),

(

(SELECT ID FROM Animals WHERE AnimalsType='PackAnimals'),

'Camels'

),

(

(SELECT ID FROM Animals WHERE AnimalsType='PackAnimals'),

'Donkeys'

);

SELECT \* FROM PackAnimals;

CREATE TABLE Dogs

(

ID int auto\_increment primary key,

PetsTypeID int not null,

Name varchar(30),

Birth date,

Commands text,

FOREIGN KEY (PetsTypeID) REFERENCES Pets(Id) ON DELETE CASCADE

);

CREATE TABLE Cats

(

ID int auto\_increment primary key,

PetsTypeID int not null,

Name varchar(30),

Birth date,

Commands text,

FOREIGN KEY (PetsTypeID) REFERENCES Pets(Id) ON DELETE CASCADE

);

CREATE TABLE Hamsters

(

ID int auto\_increment primary key,

PetsTypeID int not null,

Name varchar(30),

Birth date,

Commands text,

FOREIGN KEY (PetsTypeID) REFERENCES Pets(Id) ON DELETE CASCADE

);

CREATE TABLE Horses

(

ID int auto\_increment primary key,

PackAnTypeID int not null,

Name varchar(30),

Birth date,

Commands text,

FOREIGN KEY (PackAnTypeID) REFERENCES PackAnimals(Id) ON DELETE CASCADE

);

CREATE TABLE Camels

(

ID int auto\_increment primary key,

PackAnTypeID int not null,

Name varchar(30),

Birth date,

Commands text,

FOREIGN KEY (PackAnTypeID) REFERENCES PackAnimals(Id) ON DELETE CASCADE

);

CREATE TABLE Donkeys

(

ID int auto\_increment primary key,

PackAnTypeID int not null,

Name varchar(30),

Birth date,

Commands text,

FOREIGN KEY (PackAnTypeID) REFERENCES PackAnimals(Id) ON DELETE CASCADE

);

-- task 9

INSERT INTO Dogs(PetsTypeID, Name, Birth, Commands)

VALUES (

(SELECT ID FROM Pets WHERE PetsType='Dogs'),

'Barry',

'2022-09-10',

'Come, Stay, Lay, Follow, Catch'

),

(

(SELECT ID FROM Pets WHERE PetsType='Dogs'),

'Moska',

'2021-06-15',

'Stay, Follow, Stop, Go'

),

(

(SELECT ID FROM Pets WHERE PetsType='Dogs'),

'Jhonn',

'2016-02-01',

'He know nothing'

);

SELECT \* FROM Dogs;

INSERT INTO Cats(PetsTypeID, Name, Birth, Commands)

VALUES (

(SELECT ID FROM Pets WHERE PetsType='Cats'),

'Tom',

'2020-07-19',

'Lay on the sun, Eat'

),

(

(SELECT ID FROM Pets WHERE PetsType='Cats'),

'Borsch',

'2018-01-22',

null

),

(

(SELECT ID FROM Pets WHERE PetsType='Cats'),

'Baron',

'2023-01-01',

'Peee evrywhere'

);

SELECT \* FROM Cats;

INSERT INTO Hamsters(PetsTypeID, Name, Birth, Commands)

VALUES (

(SELECT ID FROM Pets WHERE PetsType='Hamsters'),

'Garry',

'2018-08-29',

'Eat'

),

(

(SELECT ID FROM Pets WHERE PetsType='Hamsters'),

'Larry',

'2023-02-02',

null

),

(

(SELECT ID FROM Pets WHERE PetsType='Hamsters'),

'Kuska',

'2020-12-01',

'Run'

);

SELECT \* FROM Hamsters;

INSERT INTO Horses(PackAnTypeID, Name, Birth, Commands)

VALUES (

(SELECT ID FROM PackAnimals WHERE PackAnimalsType='Horses'),

'Plotva',

'2011-08-20',

'Prrrrr, Lets go'

),

(

(SELECT ID FROM PackAnimals WHERE PackAnimalsType='Horses'),

'Wind',

'2013-04-10',

'Prrrrr, Lets go'

);

SELECT \* FROM Horses;

INSERT INTO Camels(PackAnTypeID, Name, Birth, Commands)

VALUES (

(SELECT ID FROM PackAnimals WHERE PackAnimalsType='Camels'),

'Zorg',

'2019-03-09',

'Prrrrr, Lets go'

);

SELECT \* FROM Camels;

INSERT INTO Donkeys(PackAnTypeID, Name, Birth, Commands)

VALUES (

(SELECT ID FROM PackAnimals WHERE PackAnimalsType='Donkeys'),

'Don',

'2011-06-20',

null

),

(

(SELECT ID FROM PackAnimals WHERE PackAnimalsType='Donkeys'),

'Jerry',

'2017-04-11',

null

);

SELECT \* FROM Donkeys;

-- task 10

TRUNCATE Camels;

CREATE TABLE NewPackAnimals AS

SELECT \* FROM Horses

UNION

SELECT \* FROM Donkeys;

SELECT \* FROM NewPackAnimals;

-- task 11

CREATE TABLE YoungAnimals AS

SELECT \*, TIMESTAMPDIFF(MONTH, Birth, CURDATE()) AS Age FROM Dogs

WHERE ((Birth < CURDATE() - INTERVAL 12 MONTH) AND (Birth > CURDATE() - INTERVAL 36 MONTH))

UNION

SELECT \*, TIMESTAMPDIFF(MONTH, Birth, CURDATE()) AS Age FROM Cats

WHERE ((Birth < CURDATE() - INTERVAL 12 MONTH) AND (Birth > CURDATE() - INTERVAL 36 MONTH))

UNION

SELECT \*, TIMESTAMPDIFF(MONTH, Birth, CURDATE()) AS Age FROM Hamsters

WHERE ((Birth < CURDATE() - INTERVAL 12 MONTH) AND (Birth > CURDATE() - INTERVAL 36 MONTH))

UNION

SELECT \*, TIMESTAMPDIFF(MONTH, Birth, CURDATE()) AS Age FROM NewPackAnimals

WHERE ((Birth < CURDATE() - INTERVAL 12 MONTH) AND (Birth > CURDATE() - INTERVAL 36 MONTH));

SELECT \* FROM YoungAnimals;

-- task 12

CREATE TABLE AllAnimals AS

SELECT \* FROM Dogs

UNION

SELECT \* FROM Cats

UNION

SELECT \* FROM Hamsters

UNION

SELECT \* FROM NewPackAnimals;

SELECT \* FROM AllAnimals;

13. Создать класс с Инкапсуляцией методов и наследованием по диаграмме.

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

14.1 Завести новое животное

14.2 определять животное в правильный класс

14.3 увидеть список команд, которое выполняет животное

14.4 обучить животное новым командам

14.5 Реализовать навигацию по меню

15. Создайте класс Счетчик, у которого есть метод add(), увеличивающий̆ значение внутренней̆ int переменной̆ на 1 при нажатии “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 заведении животного заполнены все поля.