

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

```
echo "Собаки" > "Домашние животные"
echo "Кошки" >> "Домашние животные"
echo "Хомяки" >> "Домашние животные"
echo "Лошади" > "Выючные животные"
echo "Верблюды" >> "Выючные животные"
echo "Ослы" >> "Выючные животные"
cat "Домашние животные" "Выючные животные" > "Друзья человека"
cat "Друзья человека"
```

2. Создать директорию, переместить файл туда.

```
mkdir Мои_животные
mv "Друзья человека" Мои_животные/
```

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

```
wget https://dev.mysql.com/get/mysql-apt-config_0.8.17-1_all.deb
sudo dpkg -i mysql-apt-config_0.8.17-1_all.deb
sudo apt update
sudo apt install mysql-server
sudo mysql_secure_installation
```

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

```
sudo dpkg -i htop.deb
sudo dpkg -r htop
sudo dpkg -P htop
```

5. Выложить историю команд в терминале ubuntu

Смотреть приложение.

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

```
-----
|   Родительский   |
|   класс           |
```



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

```
CREATE DATABASE Друзья_человека;
```

```
SHOW DATABASES;
```

```
exit;
```

8. Создать таблицы с иерархией из диаграммы в БД

```
CREATE TABLE Домашние_животные (
    id INT AUTO_INCREMENT PRIMARY KEY,
    имя VARCHAR(255) NOT NULL
);
```

Создание таблиц для подклассов "Собаки", "Кошки" и "Хомяки":

```
CREATE TABLE Собаки (
    id INT AUTO_INCREMENT PRIMARY KEY,
    порода VARCHAR(255) NOT NULL,
    FOREIGN KEY (id) REFERENCES Домашние_животные (id)
);
```

```
CREATE TABLE Кошки (
    id INT AUTO_INCREMENT PRIMARY KEY,
    окрас VARCHAR(255) NOT NULL,
    FOREIGN KEY (id) REFERENCES Домашние_животные (id)
```

);

CREATE TABLE Хомяки (

id INT AUTO_INCREMENT PRIMARY KEY,

цвет VARCHAR(255) NOT NULL,

FOREIGN KEY (id) REFERENCES Домашние_животные (id)

);

Создание таблицы "Вьючные животные":

CREATE TABLE Вьючные_животные (

id INT AUTO_INCREMENT PRIMARY KEY,

имя VARCHAR(255) NOT NULL

);

Создание таблиц для подклассов "Лошади", "Верблюды" и "Ослы":

CREATE TABLE Лошади (

id INT AUTO_INCREMENT PRIMARY KEY,

цвет VARCHAR(255) NOT NULL,

FOREIGN KEY (id) REFERENCES Вьючные_животные (id)

);

CREATE TABLE Верблюды (

id INT AUTO_INCREMENT PRIMARY KEY,

горб BOOLEAN NOT NULL,

FOREIGN KEY (id) REFERENCES Вьючные_животные (id)

);

CREATE TABLE Ослы (

id INT AUTO_INCREMENT PRIMARY KEY,

упряжка BOOLEAN NOT NULL,

FOREIGN KEY (id) REFERENCES Вьючные_животные (id)

);

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

INSERT INTO Собаки (порода, имя, команда, дата_рождения)

VALUES

('Такса', 'Барон', 'Сидеть', '2018-05-15'),
('Бигль', 'Макс', 'Лежать', '2019-02-20'),
('Овчарка', 'Рекс', 'Апорт', '2020-09-10');

Таблица "Кошки":

INSERT INTO Кошки (окрас, имя, команда, дата_рождения)

VALUES

('Рыжий', 'Мурзик', 'Играть', '2017-11-25'),
('Черный', 'Барсик', 'Мурлыкать', '2019-07-30'),
('Белый', 'Снежок', 'Ловить мышей', '2020-03-12');

Таблица "Хомяки":

INSERT INTO Хомяки (цвет, имя, команда, дата_рождения)

VALUES

('Серый', 'Чип', 'Крутить колесо', '2021-01-05'),
('Белый', 'Дейл', 'Кушать семечки', '2021-03-17');

Таблица "Лошади":

INSERT INTO Лошади (цвет, имя, команда, дата_рождения)

VALUES

('Рыжая', 'Марго', 'Верхом', '2015-08-10'),
('Черная', 'Грейс', 'Пасти траву', '2016-04-22'),
('Белая', 'Снежка', 'Верхом', '2018-01-15');

Таблица "Верблюды":

INSERT INTO Верблюды (горб, имя, команда, дата_рождения)

VALUES

(TRUE, 'Камилла', 'Нести груз', '2014-12-03'),
(TRUE, 'Максимус', 'Идти по пескам', '2017-06-28');

Таблица "Ослы":

INSERT INTO Ослы (упряжка, имя, команда, дата_рождения)

VALUES

(FALSE, 'Вася', 'Тянуть груз', '2019-10-20'),
(FALSE, 'Гриша', 'Помогать на ферме', '2020-04-08');

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

Объединение таблиц "Лошади" и "Ослы" в одну таблицу "Вьючные_животные":

-- Создайте временную таблицу "Вьючные_животные"

```
CREATE TABLE Вьючные_животные AS
```

```
SELECT * FROM Лошади;
```

-- Добавьте данные из таблицы "Ослы" в "Вьючные_животные"

```
INSERT INTO Вьючные_животные SELECT * FROM Ослы;
```

-- Удалите временные таблицы "Лошади" и "Ослы"

```
DROP TABLE Лошади;
```

```
DROP TABLE Ослы;
```

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

-- Создание таблицы "молодые животные"

```
CREATE TABLE молодые_животные AS
```

```
SELECT
```

```
    id,
```

```
    имя,
```

```
    команда,
```

```
    DATEDIFF(NOW(), дата_рождения) AS возраст_в_днях,
```

```
    CONCAT(
```

```
        FLOOR(DATEDIFF(NOW(), дата_рождения) / 30),
```

```
        ' мес ',
```

```
        DATEDIFF(NOW(), дата_рождения) % 30,
```

```
        ' дней'
```

```
    ) AS возраст_в_месяцах
```

```
FROM Домашние_животные
```

```
WHERE
```

```
    DATEDIFF(NOW(), дата_рождения) >= 365 AND
```

```
    DATEDIFF(NOW(), дата_рождения) < 1095;
```

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

-- Добавляем столбец "принадлежность" для каждой из таблиц

```
ALTER TABLE Домашние_животные ADD COLUMN принадлежность VARCHAR(255) DEFAULT 'Домашние_животные';
```

```
ALTER TABLE Вьючные_животные ADD COLUMN принадлежность VARCHAR(255) DEFAULT 'Вьючные_животные';
```

```
ALTER TABLE молодые_животные ADD COLUMN принадлежность VARCHAR(255) DEFAULT 'молодые_животные';
```

Объединение всех таблиц в одну:

-- Создаем новую таблицу "Все_животные" со всеми столбцами

```
CREATE TABLE Все_животные AS
```

```
SELECT *, принадлежность FROM Домашние_животные
```

```
UNION ALL
```

```
SELECT *, принадлежность FROM Вьючные_животные
```

```
UNION ALL
```

```
SELECT *, принадлежность FROM молодые_животные;
```

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

```
class Animal {  
    private String name;  
    private int age;  
  
    public Animal(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

```
public int getAge() {  
    return age;  
}
```

```
public void setAge(int age) {  
    this.age = age;  
}
```

```
public void makeSound() {  
    // Абстрактный метод, который будет реализован в подклассах  
}  
}
```

```
// Подкласс "Домашнее животное" с наследованием от "Животное"
```

```
class Pet extends Animal {  
    private String owner;
```

```
    public Pet(String name, int age, String owner) {  
        super(name, age);  
        this.owner = owner;  
    }
```

```
    public String getOwner() {  
        return owner;  
    }
```

```
    public void setOwner(String owner) {  
        this.owner = owner;  
    }
```

```
@Override
```

```

public void makeSound() {
    System.out.println("Мур-мя"); // Пример реализации метода для домашнего животного
}
}

```

// Пример использования классов

```

public class Main {
    public static void main(String[] args) {
        Pet cat = new Pet("Барсик", 3, "Анна");
        System.out.println("Имя: " + cat.getName());
        System.out.println("Возраст: " + cat.getAge() + " года");
        System.out.println("Владелец: " + cat.getOwner());
        cat.makeSound();
    }
}

```

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(), увеличивающий значение внутренней int переменной на 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 заведении животного заполнены все поля.