

Итоговая контрольная работа

Информация о проекте

Необходимо организовать систему учета для питомника в котором живут домашние и Pack animals.

Как сдавать проект

Для сдачи проекта необходимо создать отдельный общедоступный репозиторий (Github, gitlab, 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

Операционные системы и виртуализация (Linux)

1. Использование команды cat в Linux

- Создать два текстовых файла: "Pets"(Домашние животные) и "Pack animals"(выючные животные), используя команду ``cat`` в терминале Linux. В первом файле перечислить собак, кошек и хомяков. Во втором — лошадей, верблюдов и ослов.

- Объединить содержимое этих двух файлов в один и просмотреть его содержимое.

- Переименовать получившийся файл в "Human Friends"(.).

Пример конечного вывода после команды `"ls"` :

Desktop Documents Downloads HumanFriends.txt Music PackAnimals.txt
Pets.txt Pictures Videos

2. Работа с директориями в Linux

- Создать новую директорию и переместить туда файл "Human Friends".

3. Работа с MySQL в Linux. “Установить MySQL на вашу вычислительную машину”

- Подключить дополнительный репозиторий MySQL и установить один из пакетов из этого репозитория.

4. Управление deb-пакетами

- Установить и затем удалить deb-пакет, используя команду ``dpkg``.

5. История команд в терминале Ubuntu

- Сохранить и выложить историю ваших терминальных команд в Ubuntu.

В формате: Файла с ФИО, датой сдачи, номером группы(или потока)

Объектно-ориентированное программирование

6. Диаграмма классов

- Создать диаграмму классов с родительским классом "Животные", и двумя подклассами: "Pets" и "Pack animals", в составы классов которых в случае Pets войдут классы: собаки, кошки, хомяки, а в класс Pack animals войдут: Лошади, верблюды и ослы).

Каждый тип животных будет характеризоваться (ID, имена, даты рождения, выполняемые команды и т.д)

Диаграмму можно нарисовать в любом редакторе, такими как Lucidchart, Draw.io, Microsoft Visio и других.

7. Работа с MySQL (Задача выполняется в случае успешного выполнения задачи “Работа с MySQL в Linux. “Установить MySQL на вашу машину”

7.1. После создания диаграммы классов в 6 пункте, в 7 пункте база данных "Human Friends" должна быть структурирована в соответствии с этой диаграммой. Например, можно создать таблицы, которые будут соответствовать классам "Pets" и "Pack animals", и в этих таблицах будут поля,

которые характеризуют каждый тип животных (например, имена, даты рождения, выполняемые команды и т.д.).

7.2 - В ранее подключенном MySQL создать базу данных с названием "Human Friends".

- Создать таблицы, соответствующие иерархии из вашей диаграммы классов.

- Заполнить таблицы данными о животных, их командах и датами рождения.

- Удалить записи о верблюдах и объединить таблицы лошадей и ослов.

- Создать новую таблицу для животных в возрасте от 1 до 3 лет и вычислить их возраст с точностью до месяца.

- Объединить все созданные таблицы в одну, сохраняя информацию о принадлежности к исходным таблицам.

Пример заполненной таблицы для теста:

Лист "Pets"

ID	Name	Type	BirthDate	Commands
1	Fido	Dog	2020-01-01	Sit, Stay, Fetch
2	Whiskers	Cat	2019-05-15	Sit, Pounce
3	Hammy	Hamster	2021-03-10	Roll, Hide
4	Buddy	Dog	2018-12-10	Sit, Paw, Bark
5	Smudge	Cat	2020-02-20	Sit, Pounce, Scratch
6	Peanut	Hamster	2021-08-01	Roll, Spin
7	Bella	Dog	2019-11-11	Sit, Stay, Roll

ID	Name	Type	BirthDate	Commands
8	Oliver	Cat	2020-06-30	Meow, Scratch, Jump

Лист "PackAnimals"

ID	Name	Type	BirthDate	Commands
1	Thunder	Horse	2015-07-21	Trot, Canter, Gallop
2	Sandy	Camel	2016-11-03	Walk, Carry Load
3	Eeyore	Donkey	2017-09-18	Walk, Carry Load, Bray
4	Storm	Horse	2014-05-05	Trot, Canter
5	Dune	Camel	2018-12-12	Walk, Sit
6	Burro	Donkey	2019-01-23	Walk, Bray, Kick
7	Blaze	Horse	2016-02-29	Trot, Jump, Gallop
8	Sahara	Camel	2015-08-14	Walk, Run

8. ООП и Java

- Создать иерархию классов в Java, который будет повторять диаграмму классов созданную в задаче 6(Диаграмма классов) .

9. Программа-реестр домашних животных

- Написать программу на Java, которая будет имитировать реестр домашних животных.

Должен быть реализован следующий функционал:

9.1. Добавление нового животного

- Реализовать функциональность для добавления новых животных в реестр.

Животное должно определяться в правильный класс (например, "собака", "кошка", "хомяк" и т.д.)

9.2. Список команд животного

- Вывести список команд, которые может выполнять добавленное животное (например, "сидеть", "лежать").

9.3. Обучение новым командам

- Добавить возможность обучать животных новым командам.

9.4 Вывести список животных по дате рождения

9.5. Навигация по меню

- Реализовать консольный пользовательский интерфейс с меню для навигации между вышеуказанными функциями.

10. Счетчик животных

Создать механизм, который позволяет вывести на экран общее количество созданных животных любого типа (Как домашних, так и вьючных), то есть при создании каждого нового животного счетчик увеличивается на "1".

РЕШЕНИЕ

1. Использование команды `cat` в Linux

1.1. Создание текстовых файлов с помощью команды `cat`

Используем терминал `bash`.

Создание файла `Pets.txt`:

```
cat > Pets.txt
```

После выполнения этой команды вводим данные для, затем завершаем ввод с помощью комбинации клавиш `Ctrl + D`.

Содержимое файла `Pets.txt`:

```
Dog: Fido, Bella, Buddy
Cat: Whiskers, Smudge, Oliver
Hamster: Hammy, Peanut
```

Создание файла `PackAnimals.txt`:

```
cat > PackAnimals.txt
```

Аналогично, вводим данные для файла:

```
Horse: Thunder, Storm, Blaze
Camel: Sandy, Dune, Sahara
Donkey: Eeyore, Burro
```

1.2. Объединение файлов в один

```
cat Pets.txt PackAnimals.txt > Animals.txt
```

Просмотр содержимого нового файла:

```
anb@anb-VirtualBox:~$ cat Animals.txt
Dog: Fido, Bella, Buddy
Cat: Whiskers, Smudge, Oliver
Hamster: Hammy, Peanut
Horse: Thunder, Storm, Blaze
Camel: Sandy, Dune, Sahara
Donkey: Eeyore, Burro
```

1.3. Переименование файла

Теперь переименуем получившийся файл в `HumanFriends.txt`:

```
anb@anb-VirtualBox:~$ mv Animals.txt HumanFriends.txt
```

```
anb@anb-VirtualBox:~$ ls
```

<code>all.txt</code>	<code>ip.rules</code>	<code>shared</code>	Загрузки
<code>anb</code>	<code>ls_txt</code>	<code>snap</code>	Изображения
<code>cycles.sh</code>	<code>old-docker-compose.yaml</code>	<code>std.txt</code>	Музыка
<code>docker</code>	<code>ownersort_1.sh</code>	<code>testsc</code>	Общедоступные
<code>docker-compose.yml</code>	<code>PackAnimals.txt</code>	<code>vboxsf</code>	'Рабочий стол'
<code>err.txt</code>	<code>Pets.txt</code>	Видео	Шаблоны
<code>HumanFriends.txt</code>	<code>root</code>	Документы	

2. Работа с директориями в Linux

2.1. Создание новой директории и перемещение файла

1. Создадим новую директорию:

```
mkdir new_directory
```

2. Проверим, что директория создана:

```
anb@anb-VirtualBox:~$ ls
all.txt          ip.rules         shared           Изображения
anb              ls_txt          snap            Музыка
Animals.txt      new_directory   std.txt         Общедоступные
cycles.sh        old-docker-compose.yaml testsc          'Рабочий стол'
docker           ownersort_1.sh  vboxsf         Шаблоны
docker-compose.yml PackAnimals.txt Видео
err.txt          Pets.txt        Документы
HumanFriends.txt root
```

3. Переместим файл HumanFriends.txt в эту директорию:

```
mv HumanFriends.txt new_directory/
```

Проверяем, что файл перемещен:

```
anb@anb-VirtualBox:~$ ls new_directory
```

```
HumanFriends.txt
```

3. Работа с MySQL в Linux. Установка MySQL

3.1. Подключение дополнительного репозитория и установка пакета

1. Добавляем репозиторий MySQL (если он еще не добавлен):

```
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository 'deb http://repo.mysql.com/apt/ubuntu/focal mysql-8.0'
sudo apt-get update
```

2. Установка MySQL:

```
sudo apt-get install mysql-community-server
```

3. Проверяем, что MySQL установлен и работает:

```
sudo systemctl status mysql
```

4. Устанавливаем один из дополнительных пакетов из этого репозитория:

```
root@ubuntu2404:/home/anb# sudo apt install mysql-shell -y
```

5. Проверяем установку:

```
root@ubuntu2404:/home/anb# mysqlsh
```

```

root@ubuntu2404:/home/anb# mysqlsh
Cannot set LC_ALL to locale en_US.UTF-8: No such file or directory
MySQL Shell 8.0.41

Copyright (c) 2016, 2025, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its affiliates.
Other names may be trademarks of their respective owners.

Type '\help' or '? ' for help; '\quit' to exit.
MySQL JS >

```

4. Управление deb-пакетами

4.1. Установка deb-пакета

```

root@ubuntu2404:/home/anb# wget http://archive.ubuntu.com/ubuntu/pool/main/g/gnome-calculator/gnome-calculator\_46.0-1ubuntu1\_amd64.deb

```

```

root@ubuntu2404:/home/anb# dpkg -i gnome-calculator_46.0-1ubuntu1_amd64.deb

```

4.2. Удаление deb-пакета

```

root@ubuntu2404:/home/anb# dpkg -r gnome-calculator

```

5. История команд в терминале Ubuntu

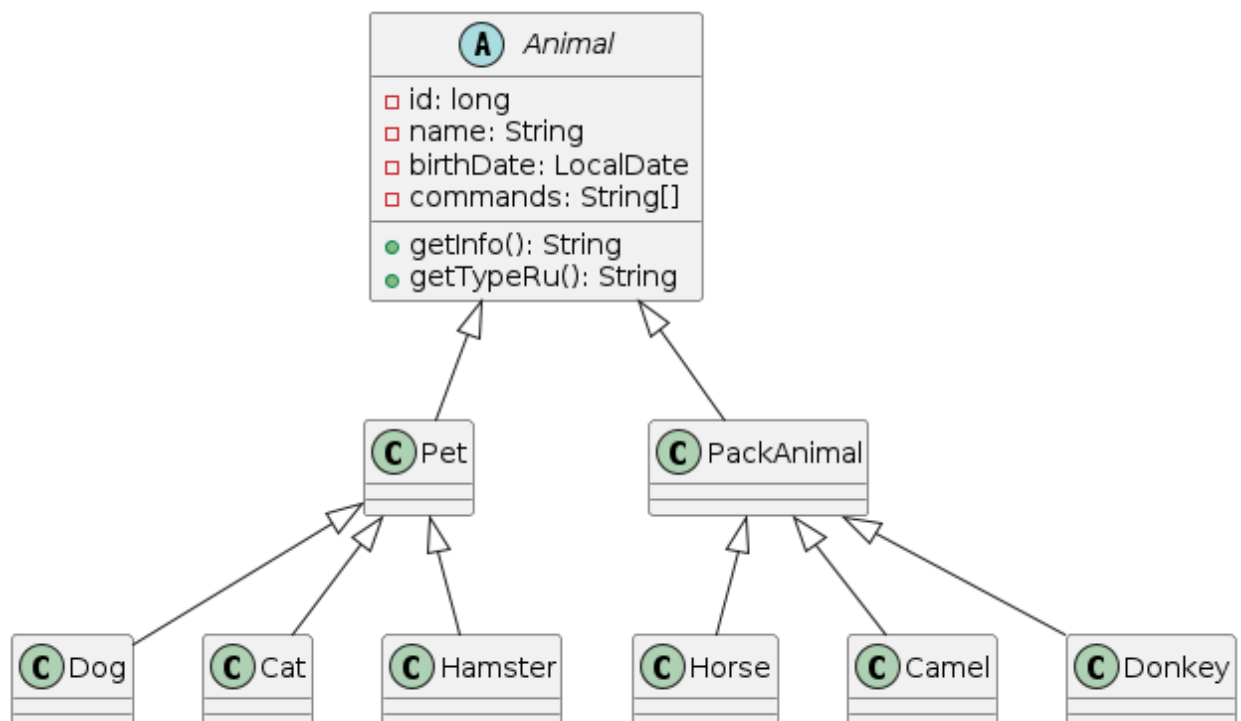
5.1. Сохранение истории команд

```

history > history_output.txt

```

6. Диаграмма классов



7. Работа с MySQL

1. Создание базы данных и таблиц:

- Создаем базу данных Human_Friends.
- Создаем таблицы для Pets и PackAnimals.

```
CREATE DATABASE Human_Friends;

CREATE TABLE Pets (
  ID INTEGER PRIMARY KEY AUTOINCREMENT,
  Name VARCHAR(50),
  Type VARCHAR(50),
  BirthDate DATE,
  Commands VARCHAR(255)
);

CREATE TABLE PackAnimals (
  ID INTEGER PRIMARY KEY AUTOINCREMENT,
  Name VARCHAR(50),
  Type VARCHAR(50),
  BirthDate DATE,
  Commands VARCHAR(255)
);
```

2. Заполнение таблиц:

```
INSERT INTO Pets (Name, Type, BirthDate, Commands) VALUES
('Fido', 'Dog', '2022-01-01', 'Sit, Stay, Fetch'),
('Whiskers', 'Cat', '2023-05-15', 'Sit, Pounce'),
('Hammy', 'Hamster', '2024-03-10', 'Roll, Hide'),
('Buddy', 'Dog', '2018-12-10', 'Sit, Paw, Bark'),
('Smudge', 'Cat', '2020-02-20', 'Sit, Pounce, Scratch'),
('Peanut', 'Hamster', '2022-08-01', 'Roll, Spin'),
('Bella', 'Dog', '2019-11-11', 'Sit, Stay, Roll'),
('Oliver', 'Cat', '2023-06-30', 'Meow, Scratch, Jump');

INSERT INTO PackAnimals (Name, Type, BirthDate, Commands) VALUES
('Thunder', 'Horse', '2015-07-21', 'Trot, Canter, Gallop'),
('Sandy', 'Camel', '2022-11-03', 'Walk, Carry Load'),
('Eeyore', 'Donkey', '2022-09-18', 'Walk, Carry Load, Bray'),
('Storm', 'Horse', '2024-05-05', 'Trot, Canter'),
('Dune', 'Camel', '2018-12-12', 'Walk, Sit'),
('Burro', 'Donkey', '2022-01-23', 'Walk, Bray, Kick'),
('Blaze', 'Horse', '2016-02-29', 'Trot, Jump, Gallop'),
('Sahara', 'Camel', '2023-08-14', 'Walk, Run');
```

3. Создание представлений по типам животных:

```
create view Dog as
  select * from Pets
  where type = 'Dog';

create view Cat as
  select * from Pets
  where type = 'Cat';

create view Hamster as
  select * from Pets
  where type = 'Hamster';

create view Horse as
  select * from Pets
  where type = 'Horse';
```

```
create view Camel as
select * from Pets
where type = 'Camel';
```

```
create view Donkey as
select * from Pets
where type = 'Donkey';
```

4. Удаление верблюдов и объединение таблиц лошадей и ослов:

```
DELETE FROM PackAnimals WHERE Type = 'Camel';
```

```
-- Объединение таблиц Лошадей и Ослов
SELECT * FROM PackAnimals WHERE Type IN ('Horse', 'Donkey');
```

5. Создание таблицы для животных в возрасте от 1 до 3 лет и вычисление их возраста:

```
CREATE VIEW AllAnimalsWithAge AS
SELECT
    ID,
    Name,
    Type,
    BirthDate,
    (CAST(strftime('%Y', 'now') AS INTEGER) - CAST(strftime('%Y', BirthDate)
AS INTEGER)) * 12 +
    (CAST(strftime('%m', 'now') AS INTEGER) - CAST(strftime('%m', BirthDate)
AS INTEGER)) AS AgeMonths,

    (CAST(strftime('%Y', 'now') AS INTEGER) - CAST(strftime('%Y', BirthDate)
AS INTEGER)) || ' г. ' ||
    ABS(CAST(strftime('%m', 'now') AS INTEGER) - CAST(strftime('%m',
BirthDate) AS INTEGER)) || ' мес.' AS Age
FROM (
    SELECT * FROM Pets
    UNION ALL
    SELECT * FROM PackAnimals
)
WHERE AgeMonths BETWEEN 12 AND 36;
```

6. Объединение всех таблиц в одну с сохранением принадлежности:

```
SELECT
    ID,
    Name,
    Type,
    BirthDate,
    'Pets' AS Source
FROM Pets

UNION ALL

SELECT
    ID,
    Name,
    Type,
    BirthDate,
    'PackAnimals' AS Source
FROM PackAnimals;
```

8. ООП и Java

1. Реализуем иерархию классов:

```
abstract class Animal {
    protected String name;
    protected String birthDate;
    protected String[] commands;

    public Animal(String name, String birthDate, String[] commands) {
        this.name = name;
        this.birthDate = birthDate;
        this.commands = commands;
    }

    abstract void displayInfo();
}

class Pet extends Animal {
    public Pet(String name, String birthDate, String[] commands) {
        super(name, birthDate, commands);
    }

    @Override
    void displayInfo() {
        System.out.println("Pet Name: " + name);
        System.out.println("Birth Date: " + birthDate);
        System.out.println("Commands: " + String.join(", ", commands));
    }
}

class PackAnimal extends Animal {
    public PackAnimal(String name, String birthDate, String[] commands) {
        super(name, birthDate, commands);
    }

    @Override
    void displayInfo() {
        System.out.println("Pack Animal Name: " + name);
        System.out.println("Birth Date: " + birthDate);
        System.out.println("Commands: " + String.join(", ", commands));
    }
}

class Dog extends Pet {
    public Dog(String name, String birthDate, String[] commands) {
        super(name, birthDate, commands);
    }
}

class Horse extends PackAnimal {
    public Horse(String name, String birthDate, String[] commands) {
        super(name, birthDate, commands);
    }
}
```

9. Программа-реестр домашних животных

<https://github.com/A-NB/Animals.git>