Истомин Сергей Андреевич   
Группа 4832

**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, 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**

**Операционные системы и виртуализация (Linux)**

1. Использование команды cat в Linux

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

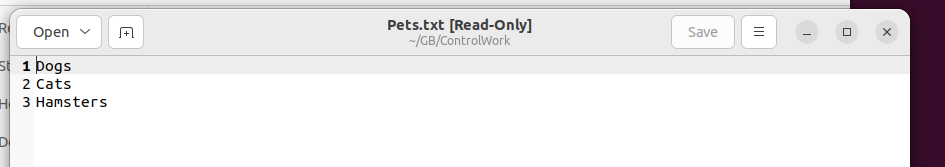
- Объединить содержимое этих двух файлов в один и просмотреть его содержимое.

- Переименовать получившийся файл в "Human Friends"(.

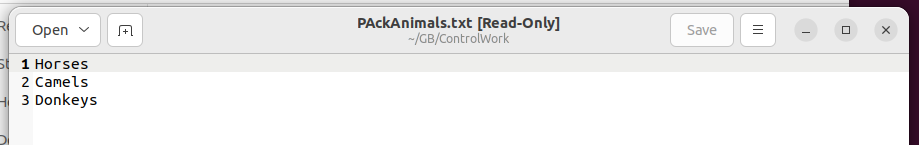
Пример конечного вывода после команды “ls” :

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

Решение:

serg@serg-VirtualBox:~$ sudo su[sudo] password for serg: root@serg-VirtualBox:/home/serg# apt updateroot@serg-VirtualBox:/home/serg# mkdir GBroot@serg-VirtualBox:/home/serg# cd GBroot@serg-VirtualBox:/home/serg/GB# mkdir ControlWorkroot@serg-VirtualBox:/home/serg/GB# cd ControlWork/root@serg-VirtualBox:/home/serg/GB/ControlWork# cat > Pets.txtDogsCatsHamsters

root@serg-VirtualBox:/home/serg/GB/ControlWork# cat > PackAnimals.txtHorsesCamelsDonkeys

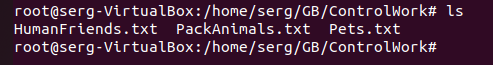


root@serg-VirtualBox:/home/serg/GB/ControlWork# cat Pets.txt PackAnimals.txt > HumanFriends.txtroot@serg-VirtualBox:/home/serg/GB/ControlWork# cat HumanFriends.txt DogsCatsHamsters

Horses

CamelsDonkeys

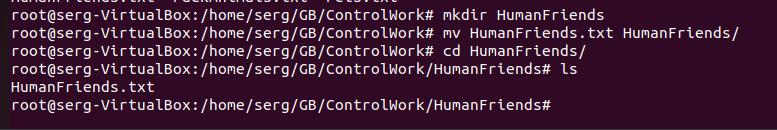
root@serg-VirtualBox:/home/serg/GB/ControlWork# lsHumanFriends.txt PackAnimals.txt Pets.txt



2. Работа с директориями в Linux

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

root@serg-VirtualBox:/home/serg/GB/ControlWork# mkdir HumanFriendsroot@serg-VirtualBox:/home/serg/GB/ControlWork# mv HumanFriends.txt HumanFriends/root@serg-VirtualBox:/home/serg/GB/ControlWork# cd HumanFriends/root@serg-VirtualBox:/home/serg/GB/ControlWork/HumanFriends# lsHumanFriends.txt



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

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

root@serg-VirtualBox:/home/serg# cd ~root@serg-VirtualBox:~# wget https://dev.mysql.com/get/mysql-apt-config\_0.8.29-1\_all.debroot@serg-VirtualBox:~# dpkg -i mysql-apt-config\_0.8.29-1\_all.deb

root@serg-VirtualBox:~# apt update

4. Управление deb-пакетами

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

root@serg-VirtualBox:~# apt update

root@serg-VirtualBox:~# apt install mysql-server

root@serg-VirtualBox:~# wget https://github.com/atom/atom/releases/download/v1.60.0/atom-amd64.deb

root@serg-VirtualBox:~# dpkg -i atom-amd64.deb

root@serg-VirtualBox:~# dpkg -r atom

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)

Выложил файл: «IstominSAdate18\_02group4832(historyLinux)»

**Объектно-ориентированное программирование**

6. Диаграмма классов

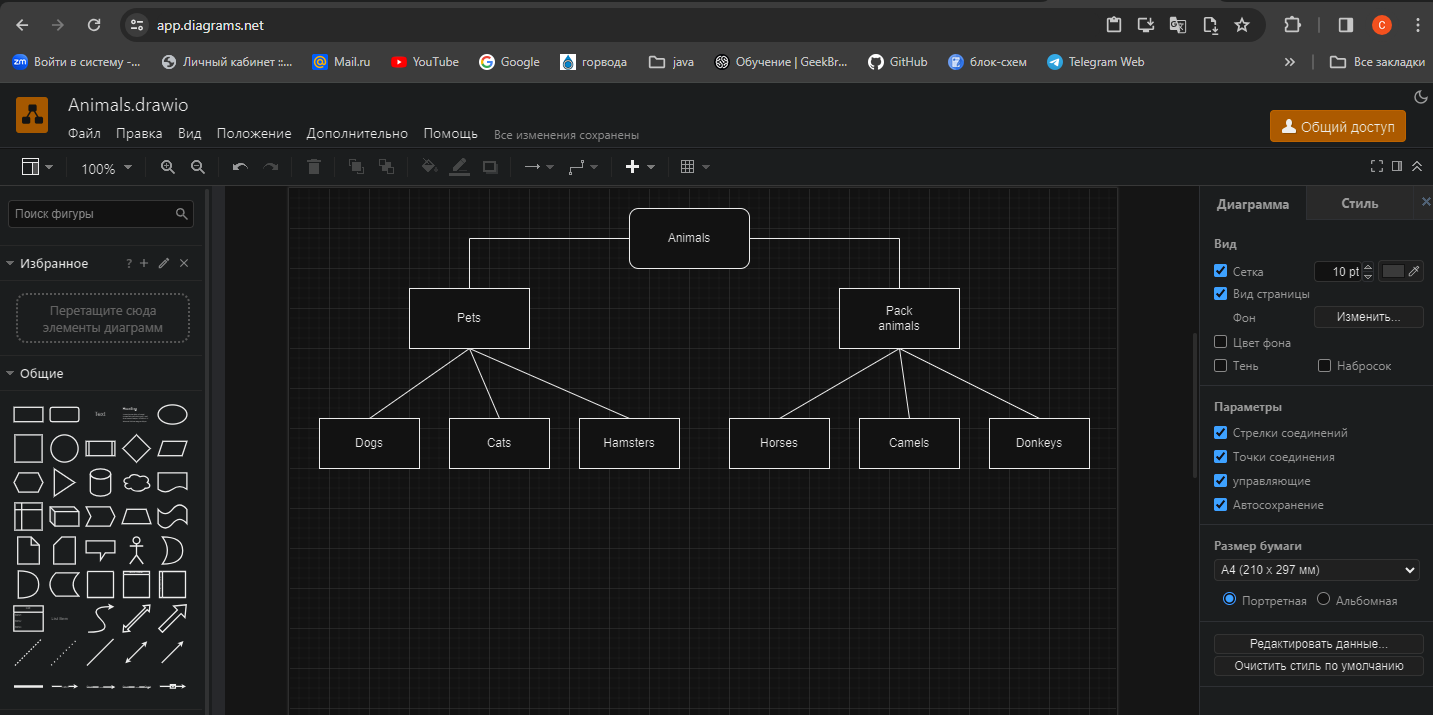
- Создать диаграмму классов с родительским классом "Животные", и двумя подклассами: "Pets" и "Pack animals".

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 (например, имена, даты рождения, выполняемые команды и т.д)

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

Файл «Animals.drawio» вложил в репозиторий



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 |
| 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 |

Решение:

-- создаем базу данных Human\_Friends

create database Human\_Friends;

-- используем базу Human\_Friends

USE Human\_Friends;

-- создаем таблицу класссов животных animal\_class (вьючные и домашние), заполняем значениями

CREATE TABLE animal\_class (ID int auto\_increment primary key, class\_name varchar(15));

INSERT INTO animal\_class (class\_name)

VALUES ('pack'),

('home');

-- создаем таблицу вьючных животных и заполняем значениями

CREATE TABLE animal\_pack (ID INT auto\_increment primary key, type varchar(15), id\_class int,

foreign key (id\_class) references animal\_class (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO animal\_pack (type, id\_class)

value ('Horse', 1),

('Donkey', 1),

('Camel', 1);

-- создаем таблицу домашних животных и заполняем значениями

CREATE TABLE animal\_home (ID INT auto\_increment primary key, type varchar(15), id\_class int,

foreign key (id\_class) references animal\_class (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO animal\_home (type, id\_class)

value ('Cat', 2),

('Dog', 2),

('Hamster', 2);

-- создаем таблицу cats и заполняем значениями

CREATE TABLE cats

(ID int AUTO\_INCREMENT PRIMARY KEY, name varchar(15), birthday date, commands varchar(30), type int,

FOREIGN KEY (type) REFERENCES animal\_home (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO cats (name, birthday, commands, type)

VALUES ('Whiskers', '2022-05-15', 'Sit, Pounce', 2),

('Smudge', '2020-02-20', "Sit, Pounce, Scratch", 2),

('Oliver', '2021-06-30', "Meow, Scratch, Jump", 2);

-- создаем таблицу dogs и заполняем значениями

CREATE TABLE dogs

(ID int AUTO\_INCREMENT PRIMARY KEY, name varchar(15), birthday date, commands varchar(30), type int,

FOREIGN KEY (type) REFERENCES animal\_home (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO dogs (name, birthday, commands, type)

VALUES ('Fido', '2020-01-01', 'Sit, Stay, Fetch', 2),

('Buddy', '2018-12-10', "Sit, Paw, Bark", 2),

('Bella', '2023-11-11', "Sit, Stay, Roll", 2);

-- создаем таблицу hamsters и заполняем значениями

CREATE TABLE hamsters

(ID int AUTO\_INCREMENT PRIMARY KEY, name varchar(15), birthday date, commands varchar(30), type int,

FOREIGN KEY (type) REFERENCES animal\_home (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO hamsters (name, birthday, commands, type)

VALUES ('Peanut', '2024-01-01', 'Roll, Spin', 2),

('Hammy', '2021-03-10', "Roll, Hide", 2);

-- создаем таблицу horses и заполняем значениями

CREATE TABLE horses

(ID int AUTO\_INCREMENT PRIMARY KEY, name varchar(15), birthday date, commands varchar(30), type int,

FOREIGN KEY (type) REFERENCES animal\_pack (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO horses (name, birthday, commands, type)

VALUES ('Thunder', '2015-07-21', 'Trot, Canter, Gallop', 1),

('Storm', '2014-05-05', "Trot, Canter", 1),

('Blaze', '2016-02-29', "Trot, Jump, Gallop", 1);

-- создаем таблицу camels и заполняем значениями

CREATE TABLE camels

(ID int AUTO\_INCREMENT PRIMARY KEY, name varchar(15), birthday date, commands varchar(30), type int,

FOREIGN KEY (type) REFERENCES animal\_pack (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO camels (name, birthday, commands, type)

VALUES ('Sandy', '2016-11-03', 'Walk, Carry Load', 1),

('Dune', '2018-12-12', "Walk, Sit", 1),

('Sahara', '2015-08-14', "Walk, Run", 1);

-- создаем таблицу donkeys и заполняем значениями

CREATE TABLE donkeys

(ID int AUTO\_INCREMENT PRIMARY KEY, name varchar(15), birthday date, commands varchar(30), type int,

FOREIGN KEY (type) REFERENCES animal\_pack (ID) ON DELETE CASCADE ON UPDATE CASCADE);

INSERT INTO donkeys (name, birthday, commands, type)

VALUES ('Burro', '2019-01-23', 'Walk, Bray, Kick', 1),

('Eeyore', '2022-12-12', "Walk, Sit", 1);

-- удалить запись о верблюдахa

TRUNCATE TABLE camels;

SELECT \* FROM camels; -- проверяем

-- объединить таблицы лошадей и ослов

SELECT id, name, birthday, commands, type FROM horses

UNION

SELECT id, name, birthday, commands, type FROM donkeys;

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

CREATE

TEMPORARY TABLE animals AS

SELECT \*, 'Horse' AS type\_animal FROM horses

UNION

SELECT \*, 'Donkey' AS type\_animal FROM donkeys

UNION

SELECT \*, 'Dog' AS type\_animal FROM dogs

UNION

SELECT \*, 'Cat' AS type\_animal FROM cats

UNION

SELECT \*, 'Hamster' AS type\_animal FROM hamsters;

CREATE TABLE yang\_animal AS

SELECT name, birthday, commands, type,

timestampdiff(MONTH, birthday, CURDATE()) AS age\_in\_month

FROM animals

WHERE birthday BETWEEN ADDDATE(curdate(), interval -3 YEAR) AND ADDDATE(CURDATE(), interval -1 YEAR);

SELECT \* FROM yang\_animal;

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

SELECT h.name, h.birthday, h.commands, pa.type, ya.age\_in\_month FROM horses h

LEFT JOIN yang\_animal ya ON ya.name = h.name

LEFT JOIN animal\_pack pa ON pa.ID = h.type

UNION

SELECT d.name, d.birthday, d.commands, pa.type, ya.age\_in\_month FROM donkeys d

LEFT JOIN yang\_animal ya ON ya.name = d.name

LEFT JOIN animal\_pack pa ON pa.ID = d.type

UNION

SELECT c.name, c.birthday, c.commands, ha.type, ya.age\_in\_month FROM cats c

LEFT JOIN yang\_animal ya ON ya.name = c.name

LEFT JOIN animal\_home ha ON ha.ID = c.type

UNION

SELECT d.name, d.birthday, d.commands, ha.type, ya.age\_in\_month FROM dogs d

LEFT JOIN yang\_animal ya ON ya.name = d.name

LEFT JOIN animal\_home ha ON ha.ID = d.type

UNION

SELECT hm.Name, hm.birthday, hm.commands, ha.type, ya.age\_in\_month FROM hamsters hm

LEFT JOIN yang\_animal ya ON ya.name = hm.name

LEFT JOIN animal\_home ha ON ha.ID = hm.type;

8. ООП и Java

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

Добавил код в папку Program.

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”.