**Итоговая контрольная работа**

**Информация о проекте**

Необходимо организовать систему учета для питомника, в котором живут домашние (Pets) и вьючные животные (Pack animals).

**Как сдавать проект**

- Для сдачи проекта необходимо создать отдельный общедоступный репозиторий ц(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 терминале операционной системы Linux**.  
   Создать два файла, Домашние животные "Pets" (заполнив файл собаками, кошками, хомяками) и Вьючные животные "Pack animals", заполнив файл (лошадьми, верблюдами и ослами), а затем объединить их. Просмотреть содержимое созданного файла. Переименовать файл, дав ему новое имя «Human Friends» (Друзья человека).
2. **Работа с директориями в Linux**.  
   Создать новую директорию и переместить туда файл "Human Friends".
3. **Работа с MySQL в Linux**.  
   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**

1. **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.
2. **Работа с MySQL** (Задача выполняется в случае успешного выполнения задачи “Работа с MySQL в Linux.)  
   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 |

1. **Иерархия в Pathon**  
   Создать иерархию классов в Pathon, который будет повторять диаграмму классов созданную в задаче 6 (Диаграмма классов)
2. **Программа-реестр домашних животных**  
   Написать программу на Python, 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
3. **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».
4. Подумать как это все засунуть в докер