

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, 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

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 осли), а затем объединить их. Просмотреть содержимое созданного файла. Переименовать файл, дав ему новое имя (Друзья человека).

2. Создать директорию, переместить файл туда.

[Необходимо создать субд MySQL, где будет храниться информация по друзьям человека](#)

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

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).

[классы по названию животного см диаграмму drawio](#)

Пункты 7-12 - это один скрипт из команд, в которых выполняются по отдельности действия в каждой задаче?

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

8. Создать таблицы с иерархией из диаграммы в БД

Вы создаете в бд таблицы, например "Лошади" и указываете там столбцы id, name и тд, и тп. Заполняете их с помощью INSERT

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

Нет, здесь нужно использовать функцию ALTER TABLE. Все задания про SQL должны быть в виде скриптов. Да, sql-файл самое оно

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

Что вы имеете в виду под "наследование по диаграмме"? - см диаграмму drawio

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(), увеличивающий значение внутренней int переменной 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.

Счетчик хранит значение в файле, например и считывает значение из файла?

Правильно понимаю, счетчик должен хранить данные отдельно от кода, в ресурсе вроде файла, к которому обращается программа?

Вообще можно просто проигнорировать try-with-resources и просто добавить возможность добавления нового животного.

Условно у Вас есть какой-то функционал и я пишу в консоль "3"(Этот пункт будет отвечать за добавление нового животного), после этого мне к примеру выдается список всех животных, чтобы я выбрал кого именно я хочу добавить, а далее я уже заполняю информацию по данному животному