**ЗАДАНИЯ ДЛЯ КУРСОВОЙ РАБОТЫ:**

1. **Напишите программу,** где класс «геометрические фигуры» (**Figure**) содержит свойство **color** с изначальным значением **white** и метод для изменения цвета фигуры, а его подклассы «овал» (**Oval**) и «квадрат» (**Square**) содержат методы \_\_**init**\_\_ для задания начальных размеров объектов при их создании.
2. **Создайте следующую программу.**

Напишите код, описывающий класс **Animal:**

* 1. Добавьте атрибут имени животного.
  2. Добавьте **метод eat()**, выводящий **“Ням ням.”**
  3. Добавьте метод **makeNoise()**, выводящий “[**animal name**] говорит **Гррр**.”
  4. Добавьте конструктор классу **Animal**, выводящий “**Родилось животное.**”

Класс **Cat:**

* 1. Пусть **Animal** будет его родительским классом.
  2. Метод **makeNoise()** класса **Cat** выводит “[**animal name**] говорит **Мяу**.”
  3. Конструктор для **Cat** выводит “**Родился кот**”, а также вызывает родительский конструктор.

Класс **Dog:**

* 1. Пусть **Animal** будет его родительским классом.
  2. Метод **makeNoise()** для **Dog** выводит “[**animal name**] говорит **Гав**.”
  3. Конструктор **Dog** выводит “**Родилась собака.**”, а также вызывает родительский конструктор.

Основная программа:

* 1. Код, создающий кота, двух собак и одно простое животное.
  2. Дайте имя каждому животному.
  3. Код, вызывающий **eat()** и **makeNoise()** для каждого животного.

1. **Напишите программу «Зооферма»** **(смотреть лекцию – презентацию),** в которой будет создано несколько объектов класса **Critter,** а манипулировать ими всеми можно будет с помощью списка. Теперь пользователь должен заботиться не об одном животном, а обо всех обитателях зоофермы. Выбирая пункт в меню, пользователь выбирает действие, которое хотел бы выполнить со всеми животными: покормить их, поиграть с ними или узнать об их самочувствии. Чтобы программа была интереснее, при создании каждого животного следует назначать ему случайно выбранные уровни голода и уныния.