

Лабораторная работа 04. Наследование и полиморфизм, отношение is-a – часть 2

Провести рефакторинг и доработку системы классов Animal-Cat-Dog из Лаб.работы 04, часть 1

1) В классе Animal

оставить только один параметризованный ведущий конструктор, всем его параметрам назначить значения по умолчанию;

добавить статическое свойство (или поле) Counter. Обеспечить подсчет в нем общего количества создаваемых животных, для этого увеличивать его в каждом вызове конструктора;

добавить абстрактный метод Sound(), в производных классах он должен возвращать в виде строки звук, издаваемый животным. Класс Animal должен стать абстрактным;

вызвать метод Sound() в методе PrintInfo

2) Добавить классы домашних и диких животных - Pet и Wild

В абстрактном классе Pet

разместить свойство PetName

определить конструктор(ы)

переопределить методы ToString, Equals, GetHashCode, метод ввода (если есть)

В абстрактном классе Wild

добавить свойство «категория опасности животного» Danger перечислимого типа, возможные значения: Unknown, White, Green, Yellow, Red, Black

определить конструктор(ы) переопределить методы ToString, Equals, GetHashCode, метод ввода (если есть)

3) Классы Cat и Dog унаследовать от Pet

соответственно перестроить их структуру

реализовать в них метод Sound

переопределить методы ToString, Equals, GetHashCode, метод ввода (если есть)

4) Добавить классы Duck, Shark, Crow, Fox.

5) Добавить классы Jet, Carlson, Submarine. Эти классы не относятся к иерархии Animal

6) Добавить интерфейсы IPrintable, IFlyable и ISwimable

Интерфейс IPrintable включает один метод PrintInfo(). Этот интерфейс реализуют классы Animal и Jet

Интерфейс ISwimable включает методы swim() и land(), у метода land должна быть реализация по умолчанию. Интерфейс реализуют классы Duck, Shark, Submarine

Интерфейс IFlyable включает

методы fly() и land(),

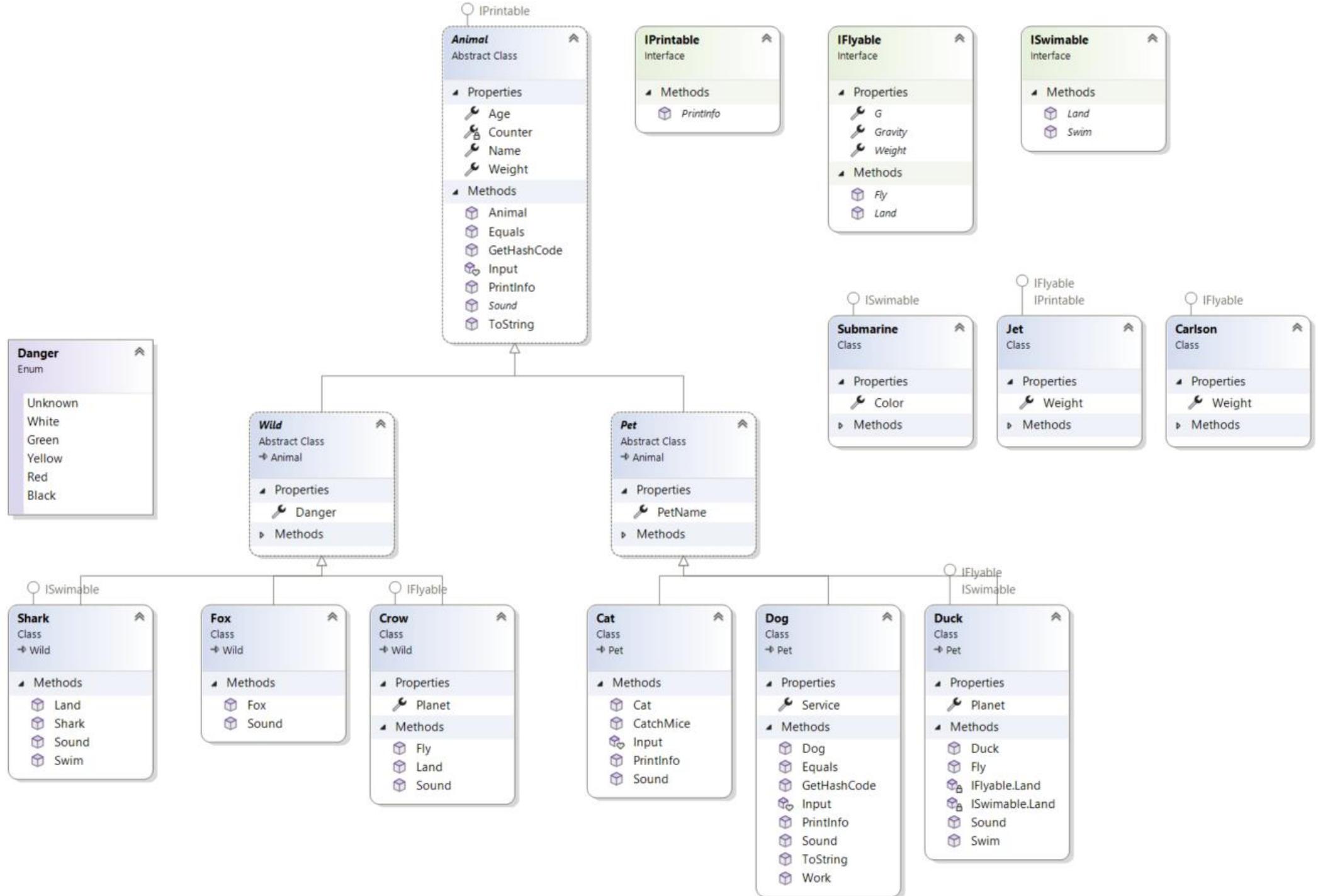
статическое свойство или константу G, по умолчанию = 9.8

метод или свойство Gravity, возвращает силу тяжести

$$F = mg = \text{Weight} * G \quad (\text{в обозначениях этого интерфейса}).$$

Его реализуют классы Duck, Crow, Jet, Carlson.

В реализациях всех этих методов в конкретных классах достаточно сделать вывод на экран какой-либо информацией, характерной для описываемого объекта.



7) В CLI-программе

Создать отдельные объекты каждого из конкретных классов, при помощи параметризованных конструкторов.

Вычислить и вывести общее количество созданных животных (статическое свойство Counter)

Создать динамические массивы типа Animal, Pet, Wild, IPrintable, IFlyable, ISwimable записать в них ссылки на те из имеющихся объектов, которые позволяют это сделать.

Для каждого из таких наборов данных выполнить характерную операцию. Например для плавающих объектов вызвать методы Swim() и Land().

Создать и протестировать метод, получающий набор летающих объектов и возвращающий тот, на который действует наименьшая сила тяжести.

Создать UML-диаграмму