Курс «Разработка на C#»

Домашняя работа №2

Студент из МФТИ, имеющий аллергию на кошек и собак, просит вас написать для него консольное приложение, которое позволило бы ему иметь виртуального домашнего питомца.

Требования к модели данных:

1. Разработать родительский класс **Animal** со следующими свойствами (Property): *Name, Age* и полем (Field) *health.*
2. В классе Animal создайте следующие методы:
   a. *public void Feed (int foodCount)* – этот метод для кормления, при его вызове необходимо увеличивать здоровье кошки (поле health) на foodCount единиц;
   b. *public void Punish (int punchCount)* – этот метод для наказания, он должен уменьшать ее здоровье на punchCount единиц.

   Иных способов изменить здоровье кошки быть не должно. Не дайте программистам, использующим ваш доменный класс, возможность написать код, который будет менять состояние здоровья минуя методы *Feed* и *Punish*).
3. Унаследовать от класса Animal следующие дочерние классы: Cat и Dog
4. В классах Cat и Dog должен присутствовать унаследованный от базового класса метод Say(), который у обоих классов выводит информацию об *Name* и *Age,* а также расширяется у каждого наследника небольшим специфичным дополнением – у кошки в дополнении ко всему выводится «Мяу», а у собаки «Гав». (примените перегрузку с обращением из перегруженного метода к функционалу базового метода + дополнения).
5. Учтите, что объектов самого типа Animal в программе быть не может, могут быть созданы только собаки и кошки (не дайте программистам возможность создать объекты класса Animal).
6. Приложение должно позволить хозяину дать имя питомцу в любой момент времени (Name), но учтите, что имя задается хозяином один единственный раз и изменению не подлежит (программисты, которые будут использовать ваш класс не должны иметь возможность написать код, который изменит уже заданное имя).
7. Проектируемый класс Animal должен требовать возраст (Age) кошки при инициализации объекта. Также спроектированный класс должен позволять узнать возраст (Age) кошки, но при этом не позволять его менять.
8. Также нужно дать возможность программистам выводить на консоль состоянии здоровья кошки. Но не в виде цифрового значения здоровья (поле *health*), а в виде «цветового» индикатора (здоровье от 0 до 50 – пишем, что окрас белый, уровень здоровья от 51 до 100 – окрас зеленый). Реализуйте это через вычисляемое свойство или метод (выбор обоснуйте).

Проверка работоспособности программы

Вам необходимо разработать консольное приложение, в котором написать статичный метод Test (static void Test(Animal pet)). Данный метод должен принимать любого питомца (не важно кошка это или собака) из объявленного и проинициализированного массива животных (Animal[] animals = new Animal[2] { new Cat(), new Dog()};) и тестировать пригодность к жизни (кормим и наказываем в цикле, подставляя различные значения в методы *Feed* и *Punish*). После тестирования всех животных из массива нужно сообщить Хозяину о состоянии здоровья путем вывода сообщения (цвет кошки можно выводить просто как «Зеленый», а можно и менять цвет вывода и выводить так «Зеленый»).