

Лабораторная работа 04. Наследование и полиморфизм, отношение is-a – часть 1

Создать класс `Animal`, включающий защищенные поля + полные свойства или автоматические свойства для работы со следующими данными:

- `Name` – название животного
- `Weight` - вес животного
- `Age` – возраст животного

В этом и всех других классах этой задачи должны быть

- не менее двух конструкторов,
- метод `ToString`,
- методы `Equals` и `GetHashCode`.

Добавить метод `PrintInfo`, который выводит все данные такого объекта на экран в подробном, отформатированном виде

На основе класса `Animal` создать два подкласса:

1) подкласс `Dog` содержит дополнительные данные:

- `PetName` – кличка
- `Service` – специальность служебной собаки (сторож, охотничья, поисковая,...)

и дополнительный метод

`void Work(int n)`

в котором ее специальность `Service` выводится на экран `n` раз, где `n` – параметр этого метода.

2) подкласс `Cat` содержит дополнительные данные

- `PetName` – кличка

и дополнительный метод

`int CatchMice()`

который возвращает случайное число в диапазоне от 0 до 50 (количество пойманных мышей)

Переопределить методы `ToString`, `Equals` и `GetHashCode` и `PrintInfo` для этих классов.

В CLI-программе

Создать отдельные объекты каждого из классов, при помощи параметризованных конструкторов; вызвать методы `PrintInfo`

Создать переменные типа `Animal`, записать в них новые объекты типов `Cat` и `Dog`

Вызвать методы `CatchMice` и `Work` от классов `Cat` и `Dog` для этих переменных .

Из всех уже созданных в программе объектов сделать один массив типа `Animal`.

Еще в этом массиве разместить три новых объекта `Animal`, `Cat`, `Dog` и ввести их с клавиатуры

Создать метод для вычисления общего веса всех животных в массиве, протестировать его.

Вывести на экран общее количество мышей, пойманных всеми кошками из этого массива.

Создать метод для поиска самого большого по весу животного. Протестировать его. Вывести на экран тип объекта, который получился в результате его вызова (кошка это или собака), если это собака, то вызвать `Work()`

Создать UML-диаграмму иерархии классов `Animal` `Cat` `Dog`

