Final Project: Tamagotchi Pet Game

Name: Khue Nhat Do

ID: 2095632

**class Pet** //a base class that provides the common attributes and functions of all pet types

--- member variables(*protected*)---

* string name //to store the name of pet
* int feed // to store percent of feeding
* int sleep //to store percent of sleep level
* int happy //to store the percent of happy level

---member functions(*public*)---

* Pet(); //default constructor that sets the name to be "Unknown", feed = 60, sleep = 60, happy = 60
* Pet(string, int, int, int); //overloaded constructor with four parameters to set name and pet’s state
* void SetName(string) ; //set the name
* string GetName(); //return the name
* int GetFeed(); //return feed level

print "Feeding...!"

feed +=20

* int GetSleep(); //return sleep level

sleep += 20

* int GetHappy(); //return happy level
* void Feed(); //make the change of feed level
* void Sleep(); //make the change of sleep level
* void Happy(); //output the message to the user that the pet is happy
* void PetState(); //print the level of pet’s state
* void PrintInfo(); //output the message to let the user know if pet’s state are lower than 20

**class Dog** : public Pet //a child class that depend on class Pet

---member functions(*public*)---

* void playingBall(); //increase happy level by 20, decrease sleep level by 20
* void NewClothes(); // increase happy level by 20
* void Next2hrs(); // decrease sleep, happy, feed levels by 20

**class Cat** : public Pet //a child class that depend on class Pet

---member functions(*public*)---

* void Rubbing() //increase happy level by 20
* void HairCut(); // increase happy level by 20
* void Next30mins(); // decrease sleep, feed levels by 30, happy level by 20

**class Hamster** : public Pet //a child class that depend on class Pet

---member functions(*public*)---

* void Running(); //increase happy level by 20, decrease sleep, feed levels by 20
* void newToys(); // increase happy level by 20
* void NextHour(); // decrease sleep, happy, feed levels by 10

Use case 1: PlayGamewithDog();

* The program shows a user menu to do action to the cat
* The user chooses the option (there will be a while loop to let the user choose different option until they want to quit)

+ Option ‘a’: see your pet's state

+ Option ‘f’: feed your pet, will change feed level

+ Option ‘s’: let your pet sleep, will change sleep level

+ Option ‘q’: quit the game

+ Option ‘h’: make your pet happy (there are two other different functions here to make your pet happy that are some unique attributes of the dog PlayBall(), NewClothes())

* Next2hrs() function will decrease sleep, happy, feed levels

Use case 2: PlayGamewithCat();

* The program shows a user menu to do action to the cat
* The user chooses the option (there will be a while loop to let the user choose different option until they want to quit)

+ Option ‘a’: see your pet's state

+ Option ‘f’: feed your pet, will change feed level

+ Option ‘s’: let your pet sleep, will change sleep level

+ Option ‘q’: quit the game

+ Option ‘h’: make your pet happy (Rubbing(), Haircut())

* Next30mins() function will decrease sleep, happy, feed levels

Use case 3: PlayGamewithHamster();

* The program shows a user menu to do action to the hamster
* The user chooses the option (there will be a while loop to let the user choose different option until they want to quit)

+ Option ‘a’: see your pet's state

+ Option ‘f’: feed your pet, will change feed level

+ Option ‘s’: let your pet sleep, will change sleep level

+ Option ‘q’: quit the game

+ Option ‘h’: make your pet happy (Running(), NewToys())

* Next1hrs() function will decrease sleep, happy, feed levels

Int main function():

+ Input the pet’s name

+ Create a file to store all information of the pet

+ Calling functions to play game