package petstore;

/\*\*

\*

\* @author Christian Dominic Angus

\*/

public interface Mammal {

public String getSound() ; // An implemented method for the Mammal

// interface that returns the string value

// of the sound the animal makes

public String getType() ; // An implemented method for the Mammal

// interface that returns the string value

// of the animal's species

public String getFood() ; // An implemented method for the Mammal

// interface that returns the string value

// of the food the animal eats

}

public abstract class Dog implements Mammal, Comparable<Mammal> {

private String myType ; // String value of the type of animal

private String mySound ; // String value of the sound the animal makes

private String myFood ; // String value of the type of food it eats

/\*\*

\* Constructor for the animal class "Dog"

\*/

public Dog() {

myType = "Dog" ;

mySound = "Woof" ;

myFood = "Omnivores" ;

}

/\*\*

\* Gives the String value of the type of animal

\* @return String value of the type of animal

\*/

public String getType() {

return myType ;

}

/\*\*

\* Gives the String value of the sound the animal makes

\* @return String value of the sound the animal makes

\*/

public String getSound() {

return mySound ;

}

/\*\*

\* Gives the String value of the food the animal eats

\* @return String value of the food the animal eats

\*/

public String getFood() {

return myFood ;

}

/\*\*

\* The abstract methods for the dog class

\*/

abstract public String getBreed() ;

abstract public String getEars() ;

abstract public String getFur() ;

abstract public String getSnout() ;

abstract public String getHeight() ;

}

public class GermanShepherd extends Dog {

private String myBreed ; // String value of the dog's breed

private String myEars ; // String value of the dog's ears

private String myFur ; // String value of the dog's fur

private String mySnout ; // String value of the dog's snout

private String myHeight ; // String value of the dog's height

/\*\*

\* A constructor for the dog class "German Shepherd"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public GermanShepherd() {

super() ;

myBreed = "German Shepherd" ;

myEars = "Pointy" ;

myFur = "Moderately fluffy" ;

mySnout = "Long" ;

myHeight = "Tall" ;

}

/\*\*

\* Gives the String value of the dog's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the dog's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the dog's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the dog's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the dog's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class Bulldog extends Dog {

private String myBreed ; // String value of the dog's breed

private String myEars ; // String value of the dog's ears

private String myFur ; // String value of the dog's fur

private String mySnout ; // String value of the dog's snout

private String myHeight ; // String value of the dog's height

/\*\*

\* A constructor for the dog class "Bulldog"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public Bulldog() {

super() ;

myBreed = "Bulldog" ;

myEars = "Floppy" ;

myFur = "Super short" ;

mySnout = "Short" ;

myHeight = "Short but wide" ;

}

/\*\*

\* Gives the String value of the dog's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the dog's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the dog's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the dog's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the dog's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class BostonTerrier extends Dog {

private String myBreed ; // String value of the dog's breed

private String myEars ; // String value of the dog's ears

private String myFur ; // String value of the dog's fur

private String mySnout ; // String value of the dog's snout

private String myHeight ; // String value of the dog's height

/\*\*

\* A constructor for the dog class "Boston Terrier"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public BostonTerrier() {

super() ;

myBreed = "Boston Terrier" ;

myEars = "Pointy" ;

myFur = "Super short" ;

mySnout = "Short" ;

myHeight = "Kinda short but has long legs" ;

}

/\*\*

\* Gives the String value of the dog's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the dog's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the dog's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the dog's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the dog's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class FrenchBulldog extends Bulldog {

private String bulldogType ; // String value for the animal's subbreed

/\*\*

\* Constructor for the bulldog class "French Bulldog"

\* It contains all of the attributes from the MiniLop super class

\* And it's own subbreed attribute

\*/

public FrenchBulldog(){

super() ;

bulldogType = "French Bulldog" ;

}

/\*\*

\* Gives the subbreed of the animal

\* @return bulldogType

\*/

public String getSubBreed() {

return bulldogType ;

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nSubbreed: "

+ getSubBreed() + "\nEars: " + getEars() + "\nFur: " + getFur()

+ "\nSnout: " + getSnout() + "\nSize: " + getHeight() ;

}

}

public abstract class Cat implements Mammal, Comparable<Mammal> {

private String myType ; // String value of the type of animal

private String mySound ; // String value of the sound the animal makes

private String myFood ; // String value of the type of food it eats

/\*\*

\* Constructor for the animal class "Cat"

\*/

public Cat() {

myType = "Cat" ;

mySound = "Meow" ;

myFood = "Carnivore";

}

/\*\*

\* Gives the String value of the type of animal

\* @return String value of the type of animal

\*/

public String getType() {

return myType ;

}

/\*\*

\* Gives the String value of the sound the animal makes

\* @return String value of the sound the animal makes

\*/

public String getSound() {

return mySound ;

}

/\*\*

\* Gives the String value of the food the animal eats

\* @return String value of the food the animal eats

\*/

public String getFood() {

return myFood ;

}

/\*\*

\* The abstract methods for the cat class

\*/

abstract public String getBreed() ;

abstract public String getEars() ;

abstract public String getFur() ;

abstract public String getSnout() ;

abstract public String getHeight() ;

}

public class Persian extends Cat {

private String myBreed ; // String value of the cat's breed

private String myEars ; // String value of the cat's ears

private String myFur ; // String value of the cat's fur

private String mySnout ; // String value of the cat's snout

private String myHeight ; // String value of the cat's height

/\*\*

\* A constructor for the dog class "Persian"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public Persian() {

super() ;

myBreed = "Persian" ;

myEars = "Pointy" ;

myFur = "Fluffy" ;

mySnout = "Short" ;

myHeight = "Short" ;

}

/\*\*

\* Gives the String value of the cat's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the cat's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the cat's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the cat's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the cat's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class Siamese extends Cat {

private String myBreed ; // String value of the cat's breed

private String myEars ; // String value of the cat's ears

private String myFur ; // String value of the cat's fur

private String mySnout ; // String value of the cat's snout

private String myHeight ; // String value of the cat's height

/\*\*

\* A constructor for the dog class "Siamese"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public Siamese() {

super() ;

myBreed = "Siamese" ;

myEars = "Pointy" ;

myFur = "Short but fluffy" ;

mySnout = "They're long for cats" ;

myHeight = "Long" ;

}

/\*\*

\* Gives the String value of the cat's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the cat's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the cat's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the cat's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the cat's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class Sphynx extends Cat {

private String myBreed ; // String value of the cat's breed

private String myEars ; // String value of the cat's ears

private String myFur ; // String value of the cat's fur

private String mySnout ; // String value of the cat's snout

private String myHeight ; // String value of the cat's height

/\*\*

\* A constructor for the dog class "Sphynx"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public Sphynx() {

super() ;

myBreed = "Sphynx" ;

myEars = "Pointy" ;

myFur = "None" ;

mySnout = "Short" ;

myHeight = "Small" ;

}

/\*\*

\* Gives the String value of the cat's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the cat's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the cat's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the cat's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the cat's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class TabbyPersian extends Persian {

private String persianType ; // String value for the animal's subbreed

/\*\*

\* Constructor for the bulldog class "French Bulldog"

\* It contains all of the attributes from the MiniLop super class

\* And it's own subbreed attribute

\*/

public TabbyPersian(){

super() ;

persianType = "Tabby Persian" ;

}

/\*\*

\* Gives the subbreed of the animal

\* @return persianType

\*/

public String getSubBreed() {

return persianType ;

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nSubbreed: "

+ getSubBreed() + "\nEars: " + getEars() + "\nFur: " + getFur()

+ "\nSnout: " + getSnout() + "\nSize: " + getHeight() ;

}

}

public abstract class Rabbit implements Mammal, Comparable<Mammal> {

private String myType ; // String value of the type of animal

private String mySound ; // String value of the sound the animal makes

private String myFood ; // String value of the type of food it eats

/\*\*

\* Constructor for the animal class "Rabbit"

\*/

public Rabbit() {

myType = "Rabbit" ;

mySound = "Rawr" ;

myFood = "Herbivore";

}

/\*\*

\* Gives the String value of the type of animal

\* @return String value of the type of animal

\*/

public String getType() {

return myType ;

}

/\*\*

\* Gives the String value of the sound the animal makes

\* @return String value of the sound the animal makes

\*/

public String getSound() {

return mySound ;

}

/\*\*

\* Gives the String value of the food the animal eats

\* @return String value of the food the animal eats

\*/

public String getFood() {

return myFood ;

}

/\*\*

\* The abstract methods for the rabbit class

\*/

abstract public String getBreed() ;

abstract public String getEars() ;

abstract public String getFur() ;

abstract public String getSnout() ;

abstract public String getHeight() ;

}

public class Dutch extends Rabbit {

private String myBreed ; // String value of the rabbit's breed

private String myEars ; // String value of the rabbit's ears

private String myFur ; // String value of the rabbit's fur

private String mySnout ; // String value of the rabbit's snout

private String myHeight ; // String value of the rabbit's height

/\*\*

\* A constructor for the dog class "Dutch"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public Dutch() {

super() ;

myBreed = "Dutch" ;

myEars = "Long and pointy" ;

myFur = "Fluffy" ;

mySnout = "Short" ;

myHeight = "Small" ;

}

/\*\*

\* Gives the String value of the rabbit's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the rabbit's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the rabbit's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the rabbit's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the rabbit's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class Harlequin extends Rabbit {

private String myBreed ; // String value of the rabbit's breed

private String myEars ; // String value of the rabbit's ears

private String myFur ; // String value of the rabbit's fur

private String mySnout ; // String value of the rabbit's snout

private String myHeight ; // String value of the rabbit's height

/\*\*

\* A constructor for the dog class "Harlequin"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public Harlequin() {

super() ;

myBreed = "Harlequin" ;

myEars = "Long and pointy" ;

myFur = "Fluffy" ;

mySnout = "Long" ;

myHeight = "Big" ;

}

/\*\*

\* Gives the String value of the rabbit's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the rabbit's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the rabbit's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the rabbit's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the rabbit's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class MiniLop extends Rabbit {

private String myBreed ; // String value of the rabbit's breed

private String myEars ; // String value of the rabbit's ears

private String myFur ; // String value of the rabbit's fur

private String mySnout ; // String value of the rabbit's snout

private String myHeight ; // String value of the rabbit's height

/\*\*

\* A constructor for the dog class "Mini Lop"

\* The constructor also contains attributes from the

\* Super class dog

\*/

public MiniLop() {

super() ;

myBreed = "MiniLop" ;

myEars = "Short and Floppy" ;

myFur = "Fluffy" ;

mySnout = "Short" ;

myHeight = "Small" ;

}

/\*\*

\* Gives the String value of the rabbit's breed

\* @return myBreed

\*/

@Override

public String getBreed() {

return myBreed ;

}

/\*\*

\* Gives the string value of the rabbit's ears

\* @return myEars

\*/

@Override

public String getEars() {

return myEars ;

}

/\*\*

\* Gives the string value of the rabbit's fur

\* @return myFur

\*/

@Override

public String getFur() {

return myFur ;

}

/\*\*

\* Gives the string value of the rabbit's snout

\* @return mySnout

\*/

@Override

public String getSnout() {

return mySnout ;

}

/\*\*

\* Gives the string value of the rabbit's height

\* @return myHeight

\*/

@Override

public String getHeight() {

return myHeight ;

}

@Override

public int compareTo(Mammal t) {

if(getType() == t.getType()) {

return 0 ;

}

else{

return 1 ;

}

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nEars: " + getEars()

+ "\nFur: " + getFur() + "\nSnout: " + getSnout() + "\nSize: "

+ getHeight() ;

}

}

public class Chinchilla extends MiniLop {

private String lopType ; // String value for the animal's subbreed

/\*\*

\* Constructor for the minilop class "Chinchilla"

\* It contains all of the attributes from the MiniLop super class

\* And it's own subbreed attribute

\*/

public Chinchilla(){

super() ;

lopType = "Chinchilla" ;

}

/\*\*

\* Gives the subbreed of the animal

\* @return lopType

\*/

public String getSubBreed() {

return lopType ;

}

@Override

public String toString() {

return "Animal: " + getType() + "\nSound: " + getSound() + "\nDiet: "

+ getFood() + "\nBreed: " + getBreed() + "\nSubbreed: "

+ getSubBreed() + "\nEars: " + getEars() + "\nFur: " + getFur()

+ "\nSnout: " + getSnout() + "\nSize: " + getHeight() ;

}

}

public class Tester {

public static void main(String[] args) {

ArrayList<Mammal> petStore = new ArrayList<Mammal>() ;

// An ArrayList to store all the animals in the pet store

/\*\*

\* Creates and stores the listed animals

\* into the petStore ArrayList

\*/

petStore.add(new Bulldog()) ;

petStore.add(new BostonTerrier()) ;

petStore.add(new GermanShepherd()) ;

petStore.add(new FrenchBulldog()) ;

petStore.add(new Persian()) ;

petStore.add(new Siamese()) ;

petStore.add(new Sphynx()) ;

petStore.add(new TabbyPersian()) ;

petStore.add(new MiniLop()) ;

petStore.add(new Dutch()) ;

petStore.add(new Harlequin()) ;

petStore.add(new Chinchilla()) ;

Scanner sc = new Scanner(System.in) ;

System.out.println("Welcome to the Pet Store!\nIf you'd like to"

+ " see what pets we have press\n1 for Dogs\n2 for Cats\n3"

+ " for Rabbit\n0 to exit the store");

int input = sc.nextInt() ;

do {

if(input > 3) {

System.out.println("Please select an option.");

input = sc.nextInt() ;

} else {

for(int i = 0 ; i < petStore.size() ; i++) {

if(input == 1 &&

petStore.get(i).getType().compareTo("Dog") == 0) {

System.out.println(petStore.get(i).toString() + "\n") ;

}

if(input == 2 &&

petStore.get(i).getType().compareTo("Cat") == 0) {

System.out.println(petStore.get(i).toString() + "\n") ;

}

if(input == 3 &&

petStore.get(i).getType().compareTo("Rabbit") == 0) {

System.out.println(petStore.get(i).toString() + "\n") ;

}

}

input = sc.nextInt() ;

}

} while(input != 0) ;

}

}