Object Oriented Programming - Assignment 3

Assignment description:

This assignment demonstrates inheritance. The Bird and Fish classes inherit from the Animal class. The Canary and Ostrich class inherit from the Bird class. The Shark and Trout inherit from the Fish class. We pass through the properties from our super class to prevent duplication of our code.

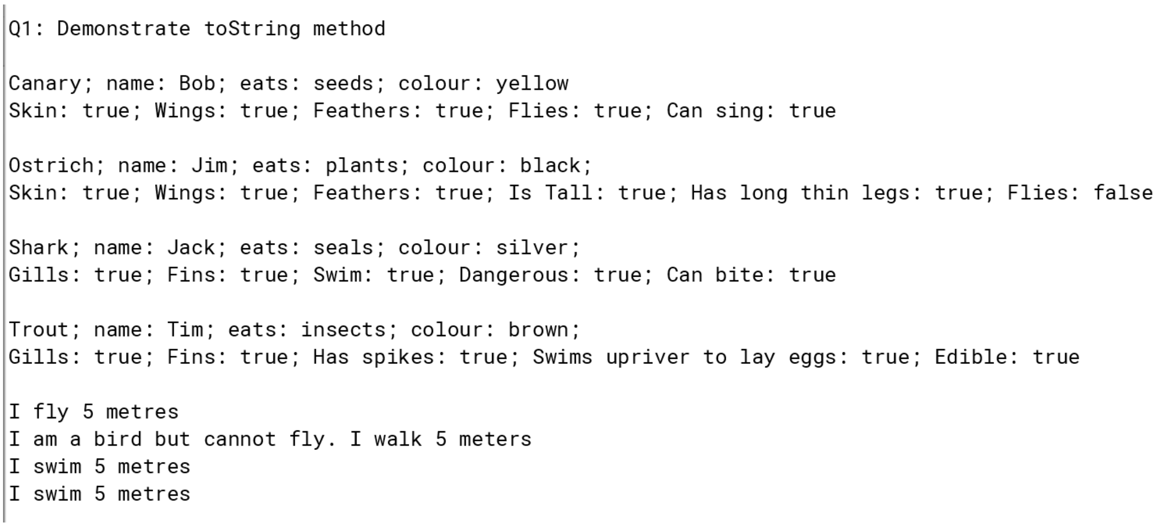
In test1 we demonstrate that our toString() methods works for each class, printing out their features by using arrays and a loop to go through each index.

In test2 we show that our equals to method works. We use an array to create new objects and a for loop to loop through comparing each element and then printing out the elements that equal each other.

Brief explanation – Ostrich:

In order to print out that the ostrich cannot fly we look at our flies method. In our Animal class we create the abstract class move(). In the bird class we have to make an if, else statement that prints “fly” if flies() is true and “I cannot fly” if flies() is false for birds. We set flies() = false in the Ostrich class. In the AnimalTest we call the move() method in test1 and it prints out that the Ostrich cannot fly.

Test1 output:



```
Q1: Demonstrate toString method

Canary; name: Bob; eats: seeds; colour: yellow
Skin: true; Wings: true; Feathers: true; Flies: true; Can sing: true

Ostrich; name: Jim; eats: plants; colour: black;
Skin: true; Wings: true; Feathers: true; Is Tall: true; Has long thin legs: true; Flies: false

Shark; name: Jack; eats: seals; colour: silver;
Gills: true; Fins: true; Swim: true; Dangerous: true; Can bite: true

Trout; name: Tim; eats: insects; colour: brown;
Gills: true; Fins: true; Has spikes: true; Swims upriver to lay eggs: true; Edible: true

I fly 5 metres
I am a bird but cannot fly. I walk 5 meters
I swim 5 metres
I swim 5 metres
```

Test2 output:

```
Q2. Use an array to demonstrate the output of the different elements

Joe the Canary in position 0 in the array matched Joe the Canary in position 6 in the array

James the Ostrich in position 1 in the array matched James the Ostrich in position 4 in the array

Sally the Trout in position 3 in the array matched Sally the Trout in position 7 in the array
```

Animal Test Class:

```
public class AnimalTest
{
    public static void main(String[] args)
    {
        AnimalTest test = new AnimalTest();

        test.test1(); // calls test1
        test.test2(); // calls test2
    }
    public void test1(){
        System.out.printf("Q1: Demonstrate toString method\n\n");

        // initialise the array of animal objects
        Animal[] animals = new Animal[4];
        animals[0] = new Canary("Bob");
        animals[1] = new Ostrich("Jim");
        animals[2] = new Shark("Jack");
        animals[3] = new Trout("Tim");

        for(int i = 0; i < animals.length;i++){ // loops over the elements of the array
            System.out.print(animals[i]);
        }
        for(Animal animal: animals){ // calling the move method
            animal.move(5);
        }
    }

    public void test2(){
        System.out.printf("\nQ2. Use an array to demonstrate the output of the different 
elements\n\n");

        // Initalising array of animal objects
        Animal[] animals = new Animal[9];
        animals[0] = new Canary("Joe");
        animals[1] = new Ostrich("James");
        animals[2] = new Shark("Paul");
        animals[3] = new Trout("Sally");
        animals[4] = new Ostrich("James");
        animals[5] = new Shark("Sean");
        animals[6] = new Canary("Joe");
        animals[7] = new Trout("Sally");
        animals[8] = new Shark("Mike");

        for (int i = 0; i < animals.length; i++) { // Runs through all the animal objects to compare them
            int y = i; // allows us to effectively compare animals in position i and y
            for (y = 0; y < animals.length; y++) {
                if (animals[i].equals(animals[y]) && i != y && !(y < i)) { // Compares animal in position i between i + 1 and 9
                    System.out.printf("%s the %s in position %d in the array matched %s the %s in position %d in the array\n\n", animals[i].getName(), animals[i].getClassName(), i, animals[y].getName(), animals[y].getClassName(), y);
                }
            }
        }
    }
}
```

Canary Class:

```
public class Canary extends Bird
{
    boolean sing;

    /**
     * Constructor for objects of class Canary
     */
    public Canary(String name)
    {
        super(); // call the constructor of the superclass Bird
        this.name = name;
        colour = "yellow"; // this overrides the value inherited from Bird
        sing = true;
    }

    /**
     * Sing method overrides the sing method
     * inherited from superclass Bird
     */
    @Override // good programming practice to use @Override to denote overridden methods
    public void sing(){
        System.out.println("tweet tweet tweet");
```

```
}


public boolean canSing(){
    return sing;
}


/**
 * toString method returns a String representation of the bird
 * What superclass has Canary inherited this method from?
 */
@Override
public String toString(){
    String str ="";
    str+= "Canary; ";
    str+= "name: ";
    str+= name;
    str+= "; ";
    str+= "eats: ";
    str+= eats;
    str+= "; ";
    str+= "colour: ";
    str+= colour;
    str+= "\n";
    str+= "Skin: ";
    str+= hasSkin();
    str+= "; ";
    str+= "Wings: ";
    str+= hasWings;
    str+= "; ";
    str+= "Feathers: ";
    str+= hasFeathers;
    str+= "; ";
    str+= "Flies: ";
    str+= flies();
    str+= "; ";
    str+= "Can sing: ";
    str+= canSing();
    str+= "\n\n";

    return str;
}
  /**
   * equals method defines how equality is defined between
   * the instances of the Canary class
   * Obj
   * return true or false depending on whether the input object is
   * equal to this Canary object
   */

  @Override
  public boolean equals(Object obj){
    //TODO : You have to define an equals method for this class
    if (obj == null){
      System.out.println("Object given is null");
      return false; //default equals
  }

  if(obj instanceof Canary){ // instanceof checks whether an object is an instance of a specific class
    Canary canary = (Canary) obj;
    if(this.getName() == canary.getName() && this.getColour() == canary.getColour()){
      return true;
    }
    }


    return false;
    }
```

Ostrich Class:

```
public class Ostrich extends Bird
{
    // instance variables - replace the example below with your own
    boolean isTall;
    boolean hasLongThinLegs;



    // Constructor for objects of class Ostrich

    public Ostrich(String name)
    {
        super(); // call the constructor of the superclass Bird
        this.name = name;
        eats = "plants";
        colour = "black";
        flies = false;
        isTall = true;
        hasLongThinLegs = true;
    }

    @Override
    public boolean flies(){
        return flies;
    }

```

```
public boolean isTall(){
    return isTall;
}

public boolean hasLongThinLegs(){
    return hasLongThinLegs;
}


 // toString method returns a String representation of the bird

@Override
public String toString(){
     String str ="";
    str+= "Ostrich; ";
    str+= "name: ";
    str+= name;
    str+= "; ";
    str+= "eats: ";
    str+= eats;
    str+= "; ";
    str+= "colour: ";
    str+= colour;
    str+= "; ";
    str+= "\n";
    str+= "Skin: ";
    str+= hasSkin();
    str+= "; ";
    str+= "Wings: ";
    str+= hasWings();
    str+= "; ";
```

```
        str+= "Feathers: ";

        str+= hasFeathers();

        str+= "; ";

        str+= "Is Tall: ";

        str+= isTall();

        str+= "; ";

        str+= "Has long thin legs: ";

        str+= hasLongThinLegs();

        str+= "; ";

        str+= "Flies: ";

        str+= flies;

        str+= "\n\n";

        return str;

  }

    @Override

    public boolean equals(Object obj){

        //TODO : You have to define an equals method for this class

        if(obj == null){

            System.out.println("Object given is null");

            return false;

        }

        if(obj instanceof Ostrich){ // instanceof checks whether an object is an instance of a specific
class

            Ostrich ostrich = (Ostrich) obj;

            if(this.getName() == ostrich.getName() && this.getColour() == ostrich.getColour()){

                return true;

            }

        }

        return false; //default equals

    }

}
```

Fish Class:

```
public class Fish extends Animal
{
    boolean hasFins;
    boolean swims;
    boolean hasGills;

    /**
     * Constructor for objects of class Fish
     */
    public Fish()
    {
        super();
        eats = "shrimp";
        colour = "gold";
        hasFins = true;
        swims = true;
        hasGills = true;
    }

    @Override
    public void move(int distance){
            System.out.printf("I swim %d metres \n", distance);
    }

    public boolean hasFins(){
        return hasFins;
    }

    /**
```

```
     * 'getter' method for the hasFeathers field
     */
    public boolean hasGills(){
        return hasGills;
    }

    public boolean swims(){
        return swims;
    }
}
```

Shark Class:

```
public class Shark extends Fish
{
    boolean canBite;
    boolean isDangerous;

    /**
     * Constructor for objects of class Shark
     */
    public Shark(String name)
    {
        super(); // calls constructor of the superclass fish
        this.name = name;
        eats = "seals";
        colour = "silver";
        canBite = true;
        isDangerous = true;
    }

    public boolean hasFins(){
        return hasFins;
    }

    public boolean canBite(){
        return canBite;
    }

    public boolean isDangerous(){
        return isDangerous;
    }

    @Override
    public String toString(){
String str ="";
        str+= "Shark; ";
        str+= "name: ";
        str+= name;
        str+= "; ";
        str+= "eats: ";
        str+= eats;
        str+= "; ";
        str+= "colour: ";
        str+= colour;
        str+= "; ";
        str+= "\n";
        str+= "Gills: ";
        str+= hasGills();
        str+= "; ";
        str+= "Fins: ";
        str+= hasFins();
        str+= "; ";
        str+= "Swim: ";
        str+= swims();
        str+= "; ";
        str+= "Dangerous: ";
        str+= isDangerous();
        str+= "; ";
        str+= "Can bite: ";
        str+= canBite();
        str+= "\n\n";
        return str;
    }
    @Override
    public boolean equals(Object obj){
        if (obj == null) {
            System.out.print("Object given is NULL\n");
            return false;
        }
        if(obj instanceof Shark){ // instanceof checks whether an object is an instance of a specific class
            Shark shark = (Shark) obj;
            if (this.getName() == shark.getName() && this.getColour() == shark.getColour()) {
                return true;
            }
        }
        return false;
    }
}
```

Trout Class:

```
public class Trout extends Fish
{
    boolean hasSpikes;
    boolean isEdible;

    boolean swimsUpriverToLayEggs;


    /**

     * Constructor for objects of class Trout

     */

    public Trout(String name)

    {

        super(); // calls constructor of the superclass fish

        this.name = name;

        eats = "insects";

        colour = "brown";

        hasSpikes = true;

        isEdible = true;

        swimsUpriverToLayEggs = true;

    }


    public boolean hasSpikes(){

        return hasSpikes;

    }


    public boolean isEdible(){

        return isEdible;

    }


    public boolean swimsUpriverToLayEggs(){

        return swimsUpriverToLayEggs;

    }


     @Override

    public String toString(){
        String str ="";
        str+= "Trout; ";
        str+= "name: ";
        str+= name;
        str+= "; ";
        str+= "eats: ";
        str+= eats;
        str+= "; ";
        str+= "colour: ";
        str+= colour;
        str+= "; ";
        str+= "\n";
        str+= "Gills: ";
        str+= hasGills();
        str+= "; ";
        str+= "Fins: ";
        str+= hasFins();
        str+= "; ";
        str+= "Has spikes: ";
        str+= hasSpikes();
        str+= "; ";
        str+= "Swims upriver to lay eggs: ";
        str+= swimsUpriverToLayEggs();
        str+= "; ";
        str+= "Edible: ";
        str+= isEdible();
        str+= "\n\n";


        return str;
    }
```

```
    @Override
    public boolean equals(Object obj){
        //TODO : You have to define an equals method for this class
         if (obj == null) {
            return false; // immediately return. Method execution goes no further
        }
         if (obj instanceof Trout) { // instanceof checks whether an object is an instance of a specific class
            // Casting the object and checking all the details are the same as the existing object
            Trout trout = (Trout) obj;
            if (this.getName() == trout.getName() && this.getColour() == trout.getColour()) {
                return true;
            }
        }
        return false; //default equals
    }
}
```