Assignment 3 OOP Maria Henehan

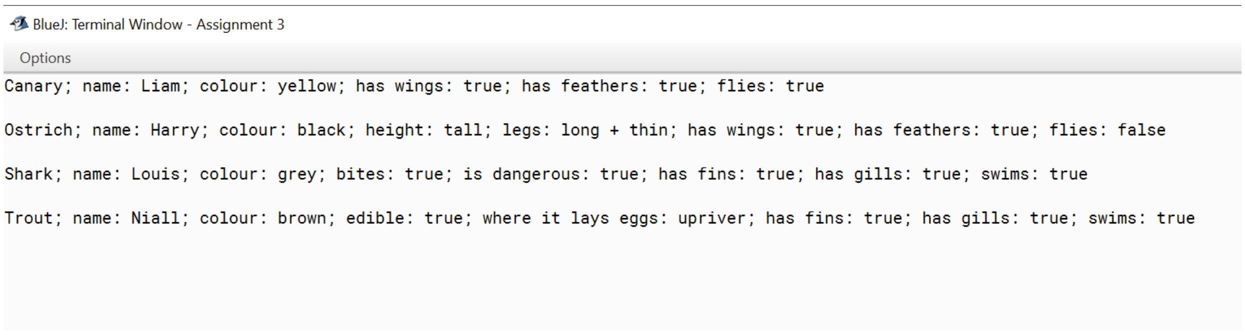
Animals inheritance assignment

• AnimalTest code

```
/**
 * Test class for the two tests
 *
 * @author (Maria Henehan)
 * @version (22/11/2020)
 */
public class testClass
{
   public static void main (String args[])
   {
     testClass test = new testClass();
     //test.test1();
     test.test2();
   }

   /**
    * test1 method
    * prints out an array of the toString() of each object
    */
   public void test1()
   {
      Animal[] animals = new Animal[4];
      animals[0] = new Canary("Liam");
      animals[1] = new Ostrich("Harry");
      animals[2] = new Shark("Louis");
      animals[3] = new Trout("Niall");
```

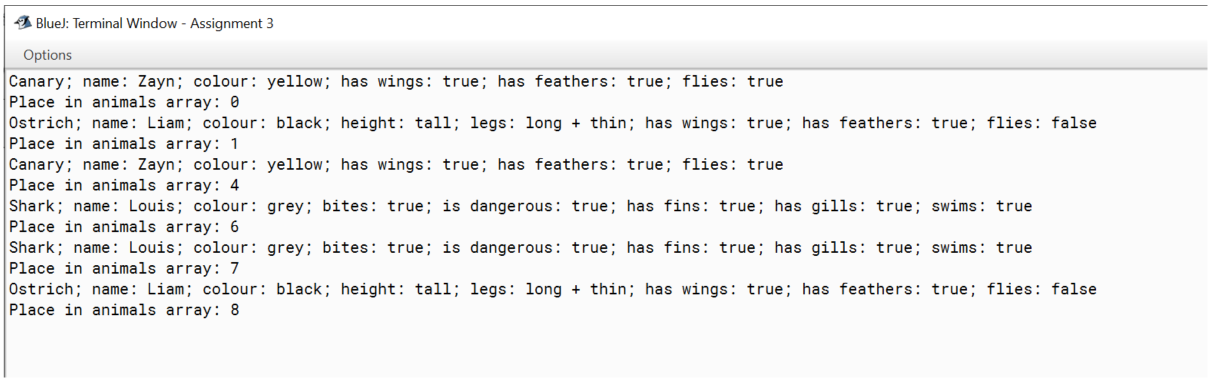
```
        for(int i=0; i<animals.length; i++)
        {
            System.out.println(animals[i]);
        }
    }

    /**
     * test2 method
     * uses nested for and if loops to check if two array entries are the same, then prints them and
their place in the array out
     */
    public void test2()
    {
        Animal[] animals = new Animal[9];
        animals[0] = new Canary("Zayn");
        animals[1] = new Ostrich("Liam");
        animals[2] = new Shark("Liam");
        animals[3] = new Trout("Niall");
        animals[4] = new Canary("Zayn");
        animals[5] = new Ostrich("Harry");
        animals[6] = new Shark("Louis");
        animals[7] = new Shark("Louis");
        animals[8] = new Ostrich("Liam");
        int i = 0;
        int j = 0;

        for(i=0; i<animals.length; i++)
        {
            for(j = animals.length -1; j>=0; j--)
            {
                if(animals[i].equals(animals[j]) && i!=j)
                {
                    System.out.print(animals[i]);
                    System.out.printf("Place in animals array: %d \n", i);
                }
            }
        }
    }
}
```

• AnimalTest test1 output



```
Canary; name: Liam; colour: yellow; has wings: true; has feathers: true; flies: true

Ostrich; name: Harry; colour: black; height: tall; legs: long + thin; has wings: true; has feathers: true; flies: false

Shark; name: Louis; colour: grey; bites: true; is dangerous: true; has fins: true; has gills: true; swims: true

Trout; name: Niall; colour: brown; edible: true; where it lays eggs: upriver; has fins: true; has gills: true; swims: true
```

• AnimalTest test2 output



```
Canary; name: Zayn; colour: yellow; has wings: true; has feathers: true; flies: true
Place in animals array: 0
Ostrich; name: Liam; colour: black; height: tall; legs: long + thin; has wings: true; has feathers: true; flies: false
Place in animals array: 1
Canary; name: Zayn; colour: yellow; has wings: true; has feathers: true; flies: true
Place in animals array: 4
Shark; name: Louis; colour: grey; bites: true; is dangerous: true; has fins: true; has gills: true; swims: true
Place in animals array: 6
Shark; name: Louis; colour: grey; bites: true; is dangerous: true; has fins: true; has gills: true; swims: true
Place in animals array: 7
Ostrich; name: Liam; colour: black; height: tall; legs: long + thin; has wings: true; has feathers: true; flies: false
Place in animals array: 8
```

• Canary code

```
/**
 * Class with fields + methods for all canaries
 * Inherits from parent class bird
 *
 * @author (Maria Henehan)
 * @version (22/11/2020)
 */
public class Canary extends Bird
{
    String name;
    /**
     * Constructor for objects of class Canary
     */
    public Canary(String name)
    {
        super();
        this.name = name;
        colour = "yellow";
    }

    @Override
    public void sing()
    {
        System.out.println("tweet tweet tweet");
    }

    /**
     * toString method
     * puts all of the attributes of a Canary object into a string and returns the string
     */
    @Override
    public String toString()
    {
        String strng ="";
        strng+= "Canary; ";
        strng+= "name: ";
        strng+= name;
        strng+= "; ";
        strng+= "colour: ";
        strng+= colour;
        strng+= "; ";
        strng+= "has wings: ";
        strng+= hasWings();
        strng+= "; ";
        strng+= "has feathers: ";
        strng+= hasFeathers;
        strng+= "; ";
        strng+= "flies: ";
        strng+= flies;
        strng+= "\n";
        // TOD0 Your job is to include the fields and attributes inherited
        //from Bird and Animal in the String representation
        return strng;
    }

    /**
     * equals method
     * checks that the object passed into the method is not null, is an instance of
     * the canary class, and if it is, checks to see if they have the same name as all
     * other canary parameters are the same
     */
    @Override
    public boolean equals(Object obj){
        //TODO : You have to define an equals method for this class
        if(obj == null)
        {
            return false;
        }
```

```
        if(obj instanceof Canary)
        {
            Canary canary = (Canary)obj;
            if(this.name == canary.name)
            {
                return true;
            }

        }
        return false; //default equals
    }

    @Override
    public void eats()
    {
        System.out.println("I eat seeds.");
    }
}
```

• Ostrich code

```
/**
 * Class with fields + methods for ostriches
 * Inherits from parent class Bird
 *
 * @author (Maria Henehan)
 * @version (22/11/2020)
 */
public class Ostrich extends Bird
{
    String name;
String height;
String legs;
/**
 * Constructor for objects of class Ostrich
 */
public Ostrich(String name)
{
  super();
  this.name = name;
  height = "tall";
  flies = false;
  legs = "long + thin";
}

/**
 * toString method
 * puts all of the attributes of an Ostrich object into a string and returns the string
 */
@Override
public String toString()
{
   String strng ="";
   strng+= "Ostrich; ";
   strng+= "name: ";
   strng+= name;
   strng+= "; ";
   strng+= "colour: ";
   strng+= colour;
   strng+= "; ";
   strng+= "height: ";
   strng+= height;
```

```
    strng+= "; ";
    strng+= "legs: ";
    strng+= legs;
    strng+= "; ";
    strng+= "has wings: ";
    strng+= hasWings();
    strng+= "; ";
    strng+= "has feathers: ";
    strng+= hasFeathers;
    strng+= "; ";
    strng+= "flies: ";
    strng+= flies;
    strng+= "\n";
    // TOD0 Your job is to include the fields and attributes inherited
    //from Bird and Animal in the String representation
    return strng;
}

/**
 * equals method
 * checks that the object passed into the method is not null, is an instance of
 * the ostrich class, and if it is, checks to see if they have the same name as all
 * other ostrich parameters are the same
 */
@Override
public boolean equals(Object obj){
    //TODO : You have to define an equals method for this class
    if(obj == null)
    {
        return false;
    }
```

```
        if(obj instanceof Ostrich)
        {
            Ostrich ostrich = (Ostrich)obj;
            if(this.name == ostrich.name)
            {
                return true;
            }

        }
        return false; //default equals
    }

    @Override
    public void eats()
    {
        System.out.println("I eat plants mainly.");
    }

    /**
     * overrides the move method to check if bird can fly before deciding which to use
     * I couldn't get it to work
     */
    @Override // good programming practice to use @Override to denote overridden methods
    public void move(int distance)
    {
        System.out.printf("I cannot fly so I walk %d metres", distance);
    }
}
```

• Fish code

```
/**
 * Abstract Fish parent class
 * Contains fields and methods for fish
 *
 * @author (Maria Henehan)
 * @version (22/11/2020)
 */
public abstract class Fish extends Animal
{
    boolean swims;
    boolean hasFins;
    boolean hasGills;
    /**
     * Constructor for objects of class Fish
     */
    public Fish()
    {
        super();
        swims = true;
        hasFins = true;
        hasGills = true;
        colour = "blue";
    }

    /**
     * overrides the move method to make it about swimming
     */
    @Override // good programming practice to use @Override to denote overridden methods
    public void move(int distance)
    {
        if(swims = true)
        {
            System.out.printf("I swim %d metres \n", distance);
        }

        else
        {
            System.out.printf("I am a fish who can't swim, so I must be dead");
        }
    }
}
```

• Shark code

```
/**
 * Class with fields and methods for Shark objects
 * Inherits from Fish parent class
 *
 * @author (Maria Henehan)
 * @version (22/11/2020)
 */
public class Shark extends Fish
{
    boolean bites;
    boolean dangerous;
    String name;
    /**
     * Constructor for objects of class Shark
     */
    public Shark(String name)
    {
        super();
        this.name = name;
        colour = "grey";
        bites = true;
        dangerous = true;
    }

    /**
     * toString method
     * puts all of the attributes of a Shark object into a string and returns the string
     */
    @Override
    public String toString()
    {
        String strng ="";
        strng+= "Shark; ";
        strng+= "name: ";
        strng+= name;
        strng+= "; ";
        strng+= "colour: ";
        strng+= colour;
        strng+= "; ";
        strng+= "bites: ";
        strng+= bites;
        strng+= "; ";
        strng+= "is dangerous: ";
        strng+= dangerous;
        strng+= "; ";
        strng+= "has fins: ";
        strng+= hasFins;
        strng+= "; ";
        strng+= "has gills: ";
        strng+= hasGills;
        strng+= "; ";
        strng+= "swims: ";
        strng+= swims;
        strng+= "\n";
        // TOD0 Your job is to include the fields and attributes inherited
        //from Bird and Animal in the String representation
        return strng;
    }

    /**
     * equals method
     * checks that the object passed into the method is not null, is an instance of
     * the shark class, and if it is, checks to see if they have the same name as all
     * other shark parameters are the same
     */
    @Override
    public boolean equals(Object obj){
        //TODO : You have to define an equals method for this class
        if(obj == null)
        {
            return false;
        }

        if(obj instanceof Shark)
        {
            Shark shark = (Shark)obj;
            if(this.name == shark.name)
            {
                return true;
            }

        }
```

```
        return false; //default equals
    }

    @Override
    public void eats()
    {
        System.out.println("I eat smaller fish.");
    }
}
```

• Trout code

```
/**
 * Class with fields and methods for objects of type Trout
 * Inherits from Fish parent class
 *
 * @author (Maria Henehan)
 * @version (22/11/2020)
 */
public class Trout extends Fish
{
    boolean edible;
    String layEggs;
    String name;
    /**
     * Constructor for objects of class Trout
     */
    public Trout(String name)
    {
        super();
        this.name = name;
        colour = "brown";
        edible = true;
    layEggs = "upriver";
}

/**
 * toString method
 * puts all of the attributes of a Trout object into a string and returns the string
 */
@Override
public String toString()
{
    String strng ="";
    strng+= "Trout; ";
    strng+= "name: ";
    strng+= name;
    strng+= "; ";
    strng+= "colour: ";
    strng+= colour;
    strng+= "; ";
    strng+= "edible: ";
    strng+= edible;
    strng+= "; ";
    strng+= "where it lays eggs: ";
    strng+= layEggs;
    strng+= "; ";
    strng+= "has fins: ";
    strng+= hasFins;
    strng+= "; ";
    strng+= "has gills: ";
    strng+= hasGills;
    strng+= "; ";
    strng+= "swims: ";
    strng+= swims;
    strng+= "\n";
    // TOD0 Your job is to include the fields and attributes inherited
    //from Bird and Animal in the String representation
    return strng;
}

/**
 * equals method
 * checks that the object passed into the method is not null, is an instance of
 * the trout class, and if it is, checks to see if they have the same name as all
 * other trout parameters are the same
 */
@Override
public boolean equals(Object obj){
    //TODO : You have to define an equals method for this class
    if(obj == null)
    {
        return false;
    }

    if(obj instanceof Trout)
    {
        Trout trout = (Trout)obj;
        if(this.name == trout.name)
        {
            return true;
        }

    }
    return false; //default equals
    }


    @Override
    public void eats()
    {
        System.out.println("I eat insects.");
    }
}
```

• Brief explanation of your code - e.g. explaining how you got the Ostrich to printout "I am a bird but cannot fly".

My code prints out "I am a bird but cannot fly" for the ostrich class but doesn't do it the way we are asked to. I attempted to do it first in the bird class using an if else statement to check if the object flies or not but this ended up only checking the flies variable in the bird class. Then I tried to do it using an if else statement in the ostrich class itself but that didn't work either. I ended up just overriding the move method completely in the ostrich class to just automatically print "I am a bird but cannot fly" for any ostriches.

I left the colour of the ostrich to be black since they are black and white or brown.

My eat method just prints out what each animal eats

I didn't include the fields in the animal method such as hasSkin and breathes in my toString statements purely because I thought they are probably assumed, obviously an animal breathes and has skin but I didn't know if we were supposed to include them

My equals methods check if the object is null first to avoid a null pointer exception and then check if the two objects are equal by first checking that they are from the same class and then by checking if they have the same name as this is the only attribute that differs between objects of the same class in my code.

My test2 checks if two elements of the animals array are the same using two for loops and one if loop. It prints out any elements that have a duplicate somewhere in the array and also prints their place in the array. It doesn't however tell you which two are the same, although that is pretty obvious from the output you get. It wouldn't be very easy to read if the array was longer but I just wanted a simple solution to check for equality without a huge amount of coding required.