Class Bird

```
java.lang.Object
  └ Bird
```

Direct Known Subclasses:

ForestBird

```
public abstract class Bird
extends java.lang.Object
```

This class encapsulates a bird and provides services to represent observations about birds observed in the wild. Each observation is assigned a timestamp. This class uses the services of the `java.util.Date` class to represent the timestamps.

Author:

mb

Constructor Summary

protected	**Bird**() Constructor for Bird.

Method Summary

Clutch	**getCurrentClutch**() This method returns the current clutch of the bird, if one is known to exist.
double	**getMeanNumberOfEggsPerClutch**() This method returns the mean number of eggs laid by this bird per clutch, as calculated over all observed clutches.
int	**getTotalNumberOfEggsLaid**() This method returns the total number of eggs laid by this bird, summing over all observed clutches.
void	**observeClutch**(int numberOfEggsObserved) Add an observation about this bird; specifically that a nest was observed that had the specified number of eggs in it.
void	**observeClutchFailure**() Add an observation about this bird; specifically that all of the eggs in this bird's current clutch have failed (e.g., all of the eggs were eaten, the entire nest fell down, or other possible reasons).
void	**observeEggFailure**() Add an observation about this bird; specifically that one of the eggs in this bird's current clutch has failed (e.g., the egg fell out of the nest, got eaten, or failed to hatch at all).
void	**observeEggHatch**() Add an observation about this bird; specifically that one of the eggs in this bird's current clutch has hatched.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Bird

```
protected Bird()
```

Constructor for Bird. This constructor cannot be invoked by a client directly and instead can only be invoked by subclasses of Bird.

Method Detail

getMeanNumberOfEggsPerClutch

`public double` **`getMeanNumberOfEggsPerClutch`**`()`

This method returns the mean number of eggs laid by this bird per clutch, as calculated over all observed clutches.

Returns:
as described above

---

observeClutch

`public void` **`observeClutch`**`(int numberOfEggsObserved)`

Add an observation about this bird; specifically that a nest was observed that had the specified number of eggs in it. This method will log the observation and attach a timestamp to it. An bird may have only one clutch at a time.

---

observeEggHatch

`public void` **`observeEggHatch`**`()`

Add an observation about this bird; specifically that one of the eggs in this bird's current clutch has hatched. This method will log the observation and attach a timestamp to it.

---

observeEggFailure

`public void` **`observeEggFailure`**`()`

Add an observation about this bird; specifically that one of the eggs in this bird's current clutch has failed (e.g., the egg fell out of the nest, got eaten, or failed to hatch at all). This method will log the observation and attach a timestamp to it.

---

observeClutchFailure

`public void` **`observeClutchFailure`**`()`

Add an observation about this bird; specifically that all of the eggs in this bird's current clutch have failed (e.g., all of the eggs were eaten, the entire nest fell down, or other possible reasons). This method will log the observation and attach a timestamp to it.

---

getTotalNumberOfEggsLaid

`public int` **`getTotalNumberOfEggsLaid`**`()`

This method returns the total number of eggs laid by this bird, summing over all observed clutches.

Returns:
as described above

---

getCurrentClutch

`public` `Clutch` **`getCurrentClutch`**`()`

This method returns the current clutch of the bird, if one is known to exist. If the bird currently does not have a clutch, this method returns null.

Returns:
as described above

Class Chickadee

```
java.lang.Object
  └ Bird
      └ ForestBird
          └ Chickadee
```

```
public class Chickadee
extends ForestBird
```

This class encapsulates an chickadee, and provides services to represent observations about specific chickadee. (e.g., such as what a naturalist would observe in the forest). It is assumed that chickadee have unique identifiers (which can be determined by the unique pattern of brown and white feathers on their chest, and that an experienced bird watcher can decode).

Author:

mb

Constructor Summary

Chickadee(java.lang.String idTag)	Construct a default representation for an chickadee.

Method Summary

void	**observePineconeEating**() Add an observation about this bird; specifically that this chickadee was seen eating seeds from a pinecone.

Methods inherited from class ForestBird

getTreeHabitat

Methods inherited from class Bird

getCurrentClutch, getMeanNumberOfEggsPerClutch, getTotalNumberOfEggsLaid, observeClutch, observeClutchFailure, observeEggFailure, observeEggHatch

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Chickadee

```
public Chickadee(java.lang.String idTag)
```

Construct a default representation for an chickadee.

Parameters:

idTag - the chickadee's unique identifier, as described in the class documentation.

Method Detail

observePineconeEating

```
public void observePineconeEating()
```

Add an observation about this bird; specifically that this chickadee was seen eating seeds from a pinecone. This method will log the observation and attach a timestamp to it.

Class Clutch

```
java.lang.Object
  └ Clutch
```

```
public class Clutch
extends java.lang.Object
```

This class encapsulates a clutch of bird eggs, which is defined as a collection of one or more eggs that were fertilized at the same time, typically laid in a single session, and incubated together. The class is defined so as to provide representation of various aspects of the clutch. Clients of this class should instantiate an empty clutch and then invoke the `addEgg(int)` method for each egg in the clutch.

Author:

mb

Constructor Summary

`Clutch()`	Constructs an empty clutch of eggs.

Method Summary

void	`addEgg()` Adds a egg to this clutch, assuming the egg has a single yolk.
void	`addEgg(int numberOfYolks)` Adds a egg to this clutch, allowing the client to indicate that the egg has more than one yolk.
Egg	`getEgg(int i)` Returns the ith egg of this clutch, thereby allowing indexed iteration over all of the eggs in the clutch.
int	`getSizeOfClutch()` Returns the size of this clutch (counted in terms of number of eggs).
int	`numberOfCalories()` Returns the total number of calories in this clutch.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Clutch

```
public Clutch()
```

Constructs an empty clutch of eggs.

Method Detail

addEgg

```
public void addEgg(int numberOfYolks)
```

Adds a egg to this clutch, allowing the client to indicate that the egg has more than one yolk.

addEgg

```
public void addEgg()
```

Adds a egg to this clutch, assuming the egg has a single yolk.

---

getEgg

```
public Egg getEgg(int i)
```

Returns the ith egg of this clutch, thereby allowing indexed iteration over all of the eggs in the clutch. The first egg has index 0. If i is less than zero or greater than the size of the clutch minus one, then this method returns null.

Returns:

as described above

---

getSizeOfClutch

```
public int getSizeOfClutch()
```

Returns the size of this clutch (counted in terms of number of eggs).

Returns:

as described above

---

numberOfCalories

```
public int numberOfCalories()
```

Returns the total number of calories in this clutch. For instance, if a preditor were to come along and eat all of the eggs, this method would provide the number of calories consumed by the preditor.

Returns:

as described above

Class Egg

```
java.lang.Object
  └ Egg
```

```
public class Egg
extends java.lang.Object
```

This class encapsulates a bird egg. The class is defined so as to provide representation of various aspects of a bird's egg (size and shape). Any eggs must exist within the context of a clutch (the lifespan of an egg is tied to the lifespan of the clutch).

Author:

mb

Method Summary

int	**getMeanLength**() This method returns the mean length of this egg (as measured by callipers, in millimeters).
double	**getMeanNumberOfYolks**() This method returns the mean number of yolks for this type of Egg (some species of birds lay eggs that have multiple yolks).
int	**getMeanWidth**() This method returns the mean width of this egg (as measured by callipers, in millimeters).

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

getMeanNumberOfYolks

```
public double getMeanNumberOfYolks()
```

This method returns the mean number of yolks for this type of Egg (some species of birds lay eggs that have multiple yolks).

Returns:

as described above

getMeanLength

```
public int getMeanLength()
```

This method returns the mean length of this egg (as measured by callipers, in millimeters).

Returns:

as described above

getMeanWidth

```
public int getMeanWidth()
```

This method returns the mean width of this egg (as measured by callipers, in millimeters).

Returns:

as described above

Class ForestBird

```
java.lang.Object
  └ Bird
      └ ForestBird
```

Direct Known Subclasses:

Chickadee, Owl

```
public class ForestBird
extends Bird
```

This class encapsulates a bird of the forest.

Author:

mb

Constructor Summary

protected	**ForestBird**() Constructor for ForestBird.

Method Summary

Tree	**getTreeHabitat**() This method returns the Tree object that encapsulates the current nesting habitat of this forest bird.

Methods inherited from class Bird
getCurrentClutch, getMeanNumberOfEggsPerClutch, getTotalNumberOfEggsLaid, observeClutch, observeClutchFailure, observeEggFailure, observeEggHatch

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ForestBird

```
protected ForestBird()
```

Constructor for ForestBird. This constructor cannot be invoked by a client directly and instead can only be invoked by subclasses of ForestBird.

Method Detail

getTreeHabitat

```
public Tree getTreeHabitat()
```

This method returns the Tree object that encapsulates the current nesting habitat of this forest bird.

Returns:

as described above

Class Maple

```
java.lang.Object
  └ Tree
      └ Maple
```

```
public class Maple
extends Tree
```

This class encapsulates a Maple tree, which is a tree of the species "Acer".

Constructor Summary

`Maple(double latitude, double longtitude)` Construct a representation for a particular maple tree, located at the specified latitude and longitude.

Method Summary

Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Maple

```
public Maple(double latitude,
             double longtitude)
```

Construct a representation for a particular maple tree, located at the specified latitude and longitude.

Parameters:

`latitude` -

`longtitude` -

Class Oak

```
java.lang.Object
  └ Tree
      └ Oak
```

```
public class Oak
extends Tree
```

This class encapsulates a Oak tree, which is a tree of the species "Quercus".

Constructor Summary

Oak(double latitude, double longtitude) Construct a representation for a particular oak tree, located at the specified latitude and longitude.

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Oak

```
public Oak(double latitude,
           double longtitude)
```

Construct a representation for a particular oak tree, located at the specified latitude and longitude.

Parameters:

latitude -

longtitude -

Class Owl

```
java.lang.Object
  └ Bird
      └ ForestBird
          └ Owl
```

```
public class Owl
extends ForestBird
```

This class encapsulates an owl, and provides services to represent observations about specific owls. (e.g., such as what a naturalist would observe in the forest). It is assumed that owls have unique identifiers (which are attached to leg bands and can be read using radio transmission).

Author:

mb

Constructor Summary

Owl(java.lang.String idTag)	Construct a default representation for an owl.

Method Summary

void	**observeMouseCatch**() Add an observation about this bird; specifically that this owl was seen catching and eating a mouse.

Methods inherited from class ForestBird

getTreeHabitat

Methods inherited from class Bird

getCurrentClutch, getMeanNumberOfEggsPerClutch, getTotalNumberOfEggsLaid, observeClutch, observeClutchFailure, observeEggFailure, observeEggHatch

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Owl

```
public Owl(java.lang.String idTag)
```

Construct a default representation for an owl.

Parameters:

idTag - the owl's unique identifier, which is attached to the owl's leg band and was read using radio transmission.

Method Detail

observeMouseCatch

```
public void observeMouseCatch()
```

Add an observation about this bird; specifically that this owl was seen catching and eating a mouse. This method will log the observation and attach a timestamp to it.

Class Tree

```
java.lang.Object
  └ Tree
```

Direct Known Subclasses:

Maple, Oak

```
public abstract class Tree
extends java.lang.Object
```

This class encapsulates a tree, which is located in a specific spot on the earth (as indicated by a particular coordinate in latitude and longitude).

Author:

mb

Constructor Summary

protected	`Tree(double latitude, double longtitude)` Constructor for Tree.

Method Summary

Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Tree

```
protected Tree(double latitude,
               double longtitude)
```

Constructor for Tree. This constructor cannot be invoked by a client directly and instead can only be invoked by subclasses of Tree.
