

# Contents

<b>Species</b>	<b>1</b>
Description . . . . .	1
Purpose . . . . .	1
Challenge . . . . .	1
Bonuses . . . . .	9
Submission . . . . .	9
Possible misconceptions . . . . .	9
Solution . . . . .	9

## Species

### Description

#### Purpose

This project is designed to teach you how to interpret and implement a simple UML specification involving multiple classes. It involves inheritance, polymorphism, properties and exception handling.

#### Challenge

**In short** Develop multiple classes to represent animal taxonomy. Different animal species have different characteristics—for example, species that belongs to the mammal class will have mammal glands—but they also all share some common properties, like their conservation status or estimated population. You need to develop at the same time a system to capture the similarities across species, and to be able to represent relevant attributes and characteristics for each species.

**In more details** We want to implement the classes pictured in the UML diagram (you right click and select “Open Image in New Tab”, or access the other versions listed in caption).

Keeping in mind that:

- Static methods are u n d e r l i n e d,
- The constructors in the `Animal` and `Mammal` classes are *protected* (that is the `#` sign): simply declare them using

```
protected Animal(string conservationStatusP, double  
↪ populationP, string wikiLinkP){...}
```

**Pay attention to details**, and note that

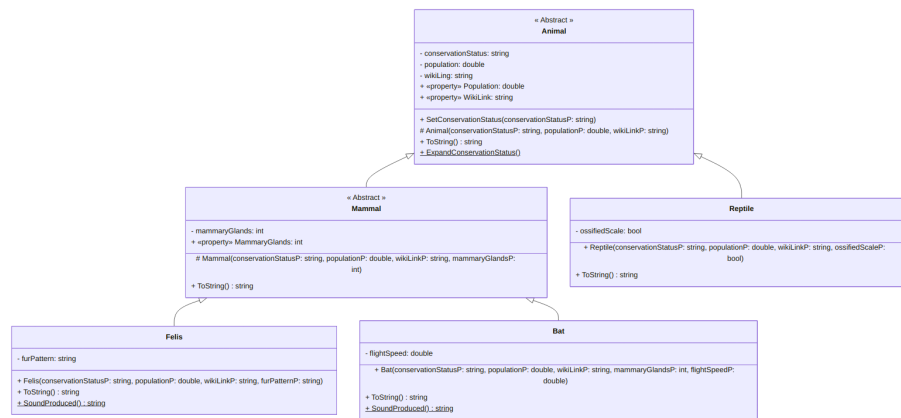


Figure 1: A UML diagram for the Animal ← Mammal class (text version, image version, svg version)

- Population is represented with a **double** as it represents *thousands* of individuals.
- The constructor for **Felis** does not require a **mammaryGlandsP** parameter, since all felis have 8 mammary glands. Bats, however, can have a varying number of mammary glands.

In addition, your code should be such that:

- The characters for the **conservationStatus** should be one of the IUCN Red List of Threatened Species code: EX, EW, CR, EN, VU, NT, CD, LC, DD, or NE. Your **SetConservationStatus** method should throw an exception if any other value is passed, and your **Animal** constructor should call the **SetConservationStatus** method.
- An exception should be thrown if **Population** is greater than 0 and **conservationStatus** is set to EX (which stands for Extinct). An exception should also be thrown for negative values, except if **conservationStatus** is set to DD (data deficient).
- An exception will be raised if the url for the **WikiLink** property does not start with <https://en.wikipedia.org/wiki/>.
- **MammaryGlands** should not accept any value not listed at [https://en.wikipedia.org/wiki/Mammary\\_gland#General](https://en.wikipedia.org/wiki/Mammary_gland#General) under "Total".
- For **Felis** and **Bat**, the **SoundProduced** methods should simply return a **string** containing "mew, meow, purr, hiss, trill, caterwaul, growl" and "screech, squeak, eek", respectively (source).
- A **Bat** flight speed cannot be negative, but it can be 0 (which is used for "unknown").
- The **ExpandConservationStatus** method should expand the abbreviations of **conservationStatus** (as given e.g., at [https://en.wikipedia.org/wiki/Conservation\\_status#IUCN\\_Red\\_List\\_of\\_Threatened\\_Species](https://en.wikipedia.org/wiki/Conservation_status#IUCN_Red_List_of_Threatened_Species)).

ned\_Species and below) and give a short definition.

- For each relevant class, the ToString method should display all the attributes and properties.

Your code should be such that the following<sup>1</sup>:

```
using System;

class Program
{
    static void Main()
    {
        // Helper string, to display more nicely
        string sep = "\n\n" + new String('*', 20) + "\n\n";

        /*
         * The following are given as indications that
         * both Animal and Mammal should be implemented as
         * abstract classes. Creating an object
         * in those classes should give you errors.
         */

        // Animal test = new Animal("EX", 12,
        ↪ "https://en.wikipedia.org/wiki/test");
        // Should return the error
        // Error CS0144: Cannot create an instance of the
        ↪ abstract type or interface 'Animal' (CS0144)
        ↪ (Species)
        // if uncommented.

        // Mammal AfricanPygmyHedgehog = new Mammal("LC",
        ↪ 10000000, "https://en.wikipedia.org/wiki/Four-
        ↪ toed_hedgehog",
        ↪ 2);
        // Should return the error
        // Error CS0144: Cannot create an instance of the
        ↪ abstract type or interface 'Mammal' (CS0144)
        ↪ (Species)
        // if uncommented.

        /*
         * First, we test the static method from the
        ↪ Animal class.
        */
    }
}
```

---

<sup>1</sup>You are welcome to test with different data and objects, of course. Just do not lose track of the important goals, which are *not* to get the right status / population estimate / wikipedia page to the species you are representing!

```

        */
        Animal.ExpandConservationStatus();
        /*
         * We now create a couple of objects from proper
↪ species.
         * The following should not return exceptions.
        */

        /*
         * We start with objects in the Felis class:
        */
        Felis JungleCat = new Felis("LC", 10,
↪ "https://en.wikipedia.org/wiki/Jungle_cat",
↪ "uniformly sandy, reddish-brown or grey fur without
↪ spots");
        Console.Write(JungleCat + sep);
        // Population is estimated, cf.
        ↪ https://www.aloki.hu/pdf/1804_58735890.pdf
        ↪ for more details.
        Felis HouseCat = new Felis("LC", 6e5,
↪ "https://en.wikipedia.org/wiki/Cat", "solid, tabby,
↪ pointed, tuxedo, calico, or tortoiseshell");
        Console.Write(HouseCat + sep);
        Felis PallasCat = new Felis("LC", 58,
↪ "https://en.wikipedia.org/wiki/Pallas%27s_cat",
↪ "light grey with black zigzags and stripes");
        Console.Write(PallasCat + sep);
        /*
         * We now have Bat objects:
        */
        Bat AnjouanMyotis = new Bat("DD", -1,
↪ "https://en.wikipedia.org/wiki/Anjouan_myotis", 2,
↪ 0);
        Console.Write(AnjouanMyotis + sep);
        Bat BlackEaredFlyingFox = new Bat("VU", .4,
↪ "https://en.wikipedia.org/wiki/Black-
↪ eared_flying_fox", 2, 22); //Flight speed given in
↪ mph
        Console.Write(BlackEaredFlyingFox + sep);
        Bat DesmodusDraculae = new Bat("EX", 0,
↪ "https://en.wikipedia.org/wiki/Desmodus_draculae", 2,
↪ 0);
        Console.Write(DesmodusDraculae + sep);

        /*

```

```

    * Finally, one Reptile object:
    */
    Reptile BallPython = new Reptile("NT", .1,
↪ "https://en.wikipedia.org/wiki/Ball_python", false);
    Console.WriteLine(BallPython + sep);
    /*
    * We now test our improper values handling.
    */
    Console.WriteLine("Test 1:");
    try
    {
        Felis test1 = new Felis("INVALID CODE", 1,
↪ null, null);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
    Console.WriteLine("Test 2:");
    try
    {
        Felis test2 = new Felis("EX", 10,
↪ "https://en.wikipedia.org/wiki/whatever", null);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
    Console.WriteLine("Test 3:");
    try
    {
        Felis test3 = new Felis("EX", 0,
↪ "http://sketchy-website.com/", null);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
    Console.WriteLine("Test 4:");
    try
    {
        Bat test4 = new Bat("LC", -10, null, 0, 12);
    }
    catch (Exception e)

```

```

    {
        Console.WriteLine(e.Message);
    }
    Console.WriteLine("Test 5:");
    try
    {
        Bat test5 = new Bat("LC", 10,
↪ "https://en.wikipedia.org/wiki/whatever", 3, 0);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
    Console.WriteLine("Test 6:");
    try
    {
        Bat test5 = new Bat("LC", 1,
↪ "https://en.wikipedia.org/wiki/whatever", 2, -1);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
}
}

```

should display (something along the lines of)

Extinct (EX) - There are no known living individuals

Extinct in the wild (EW) - Known only to survive in  
 ↪ captivity, or as a naturalized population outside its  
 ↪ historic range

Critically Endangered (CR) - Highest risk of extinction  
 ↪ in the wild

Endangered (EN) - Higher risk of extinction in the wild

Vulnerable (VU) - High risk of extinction in the wild

Near Threatened (NT) - Likely to become endangered in the  
 ↪ near future

Conservation Dependent (CD) - Low risk; is conserved to  
 ↪ prevent being near threatened, certain events may  
 ↪ lead it to being a higher risk level

Least concern (LC) - Very Low risk; does not qualify for  
 ↪ a higher risk category and not likely to be  
 ↪ threatened in the near future. Widespread and  
 ↪ abundant taxa are included in this category.

Data deficient (DD) - Not enough data to make an  
 ↪ assessment of its risk of extinction

Not evaluated (NE) - Has not yet been evaluated against  
↳ the criteria.

Population	10 000
Conservation Status	LC
Wikipedia Link	
↳	<a href="https://en.wikipedia.org/wiki/Jungle_cat">https://en.wikipedia.org/wiki/Jungle_cat</a>
Mammary Glands	8
Fur pattern	uniformly sandy, reddish-brown or
↳	grey fur without spots
Sound(s) produced	mew, meow, purr, hiss, trill,
↳	caterwaul, growl

\*\*\*\*\*

Population	600 000 000
Conservation Status	LC
Wikipedia Link	<a href="https://en.wikipedia.org/wiki/Cat">https://en.wikipedia.org/wiki/Cat</a>
Mammary Glands	8
Fur pattern	solid, tabby, pointed, tuxedo,
↳	calico, or tortoiseshell
Sound(s) produced	mew, meow, purr, hiss, trill,
↳	caterwaul, growl

\*\*\*\*\*

Population	58 000
Conservation Status	LC
Wikipedia Link	
↳	<a href="https://en.wikipedia.org/wiki/Pallas%27s_cat">https://en.wikipedia.org/wiki/Pallas%27s_cat</a>
Mammary Glands	8
Fur pattern	light grey with black zigzags and
↳	stripes
Sound(s) produced	mew, meow, purr, hiss, trill,
↳	caterwaul, growl

\*\*\*\*\*

Population	-1 000
Conservation Status	DD
Wikipedia Link	
↳	<a href="https://en.wikipedia.org/wiki/Anjouan_myotis">https://en.wikipedia.org/wiki/Anjouan_myotis</a>
Mammary Glands	2

Flight speed	0
Sound(s) produced	screech, squeak, eek

\*\*\*\*\*

Population	400
Conservation Status	VU
Wikipedia Link	
↪	<a href="https://en.wikipedia.org/wiki/Black-eared_flying_fox">https://en.wikipedia.org/wiki/Black-eared_flying_fox</a>
Mammary Glands	2
Flight speed	22
Sound(s) produced	screech, squeak, eek

\*\*\*\*\*

Population	0
Conservation Status	EX
Wikipedia Link	
↪	<a href="https://en.wikipedia.org/wiki/Desmodus_draculae">https://en.wikipedia.org/wiki/Desmodus_draculae</a>
Mammary Glands	2
Flight speed	0
Sound(s) produced	screech, squeak, eek

\*\*\*\*\*

Population	100
Conservation Status	NT
Wikipedia Link	
↪	<a href="https://en.wikipedia.org/wiki/Ball_python">https://en.wikipedia.org/wiki/Ball_python</a>
Ossified Scales	False

\*\*\*\*\*

Test 1:  
A conservation status is 2 characters long.  
Test 2:  
Population cannot be non-zero if conservation status is  
↪ not EX.  
Test 3:  
Submission link (<http://sketchy-website.com/>) does not  
↪ start with "<https://en.wikipedia.org/wiki/>".  
Test 4:



Population cannot be negative unless conservation status  
↪ is DD.

Test 5:

Number of mammary glands cannot be odd, unless it is 13,  
↪ 25 or 27.

Test 6:

Flight speed cannot be negative.

### Bonuses

This project is already challenging *as is*, but bonus will be given if:

- (easy) The `ToString()` methods return the information nicely formatted, as above.
- (medium) You explain briefly (in comment) why the `SoundProduced` method is not given as an abstract static method in the `Mammal` class.
- (hard) The `WikiLink` property accepts any “WP code” from wikipedia: links starting with `https://id.wikipedia.org/wiki/`, `https://simple.wikipedia.org/wiki/`, etc. should also be accepted.

### Submission

Please, follow our guideline on project submission. In particular, make sure you write your name and the date in a delimited comment at the beginning of your file.

### Possible misconceptions

Some additional possible misconceptions are discussed in this project.

### Solution

A possible solution is shared in this archive

Note that

- in the `Animal` class, a `EpsilonD` constant was added to test if a double is “almost” 0,
- in the `Bat` class, a `FlightSpeed` property was added to facilitate testing the validity of its value,
- in the `Bat` class, objects are created even if the number of mammary glands is not 2 or 4, but a message is displayed,

This solution completes the “(easy) The `ToString()` methods return the information nicely formatted, as above.” bonus, and a solution to the

“(medium) You explain briefly (in comment) why the SoundProduced method is not given as an abstract static method in the Mammal class.” can be as follows:

In short, static methods cannot be inherited or overridden in C#, and that is why they cannot be abstract. This is discussed in some details in this stack exchange post. A change was recently introduced in .NET 6, that makes static abstract members in interface possible, but it is limited to interfaces, and abstract static methods cannot be added to abstract classes.