Homework Assignment 3

Late homework assignments will not be accepted, unless you have a valid written excuse (medical, etc.). You must do this assignment alone. No team work or "talking with your friends" will be accepted. No copying from the Internet. Cheating means zero.

Create a new Eclipse workspace named "**`Assignment3_1234567890`**" on the desktop of your computer (replace **`1234567890`** with your student ID number). For each question below, create a new project in that workspace. Call each project by its question number: "**`Question1`**", "**`Question2`**", etc. Answer all the questions below. At the end of the assignment, create a ZIP archive of the whole workspace folder. The resulting ZIP file must be called "**`Assignment3_1234567890.zip`**" (replace **`1234567890`** with your student ID number). Upload the ZIP file on iSpace.

Here are a few extra instructions:

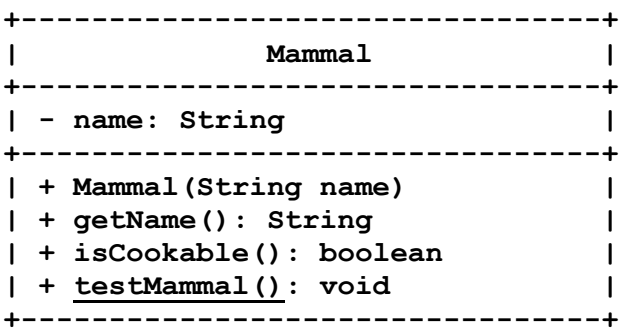
- Do not forget to write tests for *all* the code of *all* the classes.
- Give meaningful names to your variables so we can easily know what each variable is used for in your program.
- Put comments in your code (in English!) to explain WHAT your code is doing and also to explain HOW your program is doing it.
- Make sure all your code is properly indented (formatted). Your code must be beautiful to read.

Failure to follow these instructions will result in you losing points.

Due Date: **23:50 pm on Apr 17 (Sunday).**

Question 1

Create a class **`Mammal`** with the following UML diagrams:



The **`isCookable`** method of the **`Mammal`** class returns a boolean indicating whether the mammal can be cooked or not: some mammals can be cooked and some mammals cannot be cooked so for safety reasons the **`isCookable`** method just prints a message "**`Do not cook this!`**" and returns **`false`** (because the **`isCookable`** method must return a boolean).

Add a class **`Human`** to your program. A human is a mammal. The constructor for the **`Human`** class takes no argument. All humans are named "**`Alice`**" and humans cannot be cooked (the **`isCookable`** method must not print any message though, it simply returns **`false`**).

Add a class **`Rabbit`** to your program. A rabbit is a mammal. The **`Rabbit`** class has a private instance variable **`weight`** of type **`double`** that describes the weight of the rabbit, and a **`getWeight`** method. The constructor for the **`Rabbit`** class takes the rabbit's name and the rabbit's weight as arguments. A rabbit can be cooked.

Add a class **`EuropeanRabbit`** to your program. European rabbit is a species of rabbit. The **`EuropeanRabbit`** class has two constructors: the first constructor takes the European rabbit's name and the European rabbit's weight as arguments; the second constructor only takes the European rabbit's name as argument and always uses **`2.0`** as the European rabbit's weight. The second constructor must use the first constructor.

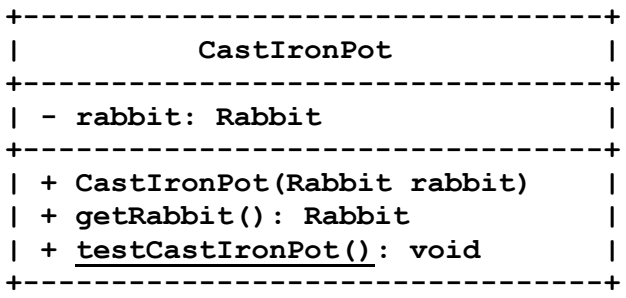
Add a class **`LapinSauteChasseur`** to your program. Lapin sauté chasseur is a kind of European rabbit. The constructor for the **`LapinSauteChasseur`** class takes no argument. Lapin sauté chasseur is always named "**`Delicious`**" and has a weight of **`0.5`**.

Add a class **`FrankTheRabbit`** to your program. Frank The Rabbit is a kind of rabbit. The constructor for the **`FrankTheRabbit`** class takes no argument. Frank The Rabbit is always named "**`Frank`**", has a weight of **`100.0`**, and cannot be cooked.

Add a class **`Start`** to your program to test all your classes.

Question 2

Add a class **`CastIronPot`** to your program with the following UML diagram:



In the **`testCastIronPot`** method, create a lapin sauté chasseur called **`lsc1`**, then create a cast iron pot with this lapin sauté chasseur in it. Then get the lapin sauté chasseur from the cast iron pot and store it into a local variable called **`lsc2`** of type **`LapinSauteChasseur`**. Use the **`==`** operator to check that **`lsc1`** and **`lsc2`** are the same lapin sauté chasseur.