MSE1122 Assignment 1 2014

Submission details:

- You may work on your own or in a group of 2 students.
- When working in a group; only submit a single assignment – otherwise it will be seen as a copied assignment.
- Your assignment must be handed in as a printed document, with the following details:
    - A cover page which contains the heading "MSE1122 Assignment 1 2014"; followed by the student number, name and surname of each group member.
    - A printout of each class file; except for the Program.cs class.
    - Provide the printout of every class with an appropriate heading, so there can be no confusion when marking.
    - Your document must have the following structure:
        - Cover page
        - Interface IBodyPart
        - Class Eye
        - Class Head
        - Class Limb
        - Class Arm
        - Class Tentacle
    - Your document must be bound or stapled. Loose pages will not be accepted.
- You must hand in your printed document by the end of your practical period (09:15) on Thursday 4 September 2014.

Specifications:

Lucas Engineering is a tech start-up company which aims to build a robot labour force for the future. They require a system which allows them to keep track of the various robotic body parts which they are developing.

Your software designers have come up with the following specifications, which you are required to turn into functional C# code:

1. Develop an interface called IBodyPart. The interface must be developed as follows:
    a. A private method, called GenerateSerial, which receives no parameters and returns no values.
2. Develop a class called Eye. The class must be developed as follows:
    a. It must make use of the interface IBodyPart.
    b. It must have a private string-based field for storing the Serial.
    c. It must have a private static integer counter, called EyeCount, which keeps track of the number of Eyes which have been created. The EyeCount must be automatically incremented whenever an Eye is created.

d. It must have an auto-implemented property, called EyeColour, for saving and returning the colour of the eye.

e. The constructor should receive the string-based colour of the Eye as its parameter and calculate the Serial of the eye.

f. An Eye's serial number should be created by using the current number of existing Eyes and combining it with the letter E. The serial number may never be less than 6 characters in length e.g. the very first Eye object should have a serial of E00001.

g. An override of the ToString() method must be provided, which uses GetType() to return the type of the object, as well as its serial number and the colour of the eye. You may use your own discretion to format the returned string.

3. Develop a class called Head. The class must be developed as follows:

   a. It must make use of the interface IBodyPart.

   b. It must contain two private Eye fields, for storing a left and a right eye.

   c. It must have a private string-based field for storing the Serial.

   d. It must have a private static integer counter, called HeadCount, which keeps track of the number of Heads which have been created. The HeadCount must be automatically incremented whenever a Head is created.

   e. The constructor must receive two string-based parameter, which are the colours for the left and right eyes. These two string values must be used to create and initiliaze the class' two Eye fields.

   f. A Head's serial number should be created by using the current number of existing Heads and combining it with the letter H. The serial number may never be less than 6 characters in length e.g. the very first Head object should have a serial of H00001.

   g. An override of the ToString() method must be provided, which uses GetType() to return the type of the object, as well as its serial number and the complete details of both the Head's Eyes. You may use your own discretion to format the returned string.

4. Develop an abstract class called Limb. The class must be developed as follows:

   a. It must make use of the interface IBodyPart.

   b. It must have a string-based field for storing the Serial, which will be accessible in its derived classes.

   c. It must have a private static integer counter, called LimbCount, which keeps track of the number of Limbs which have been created. The LimbCount must be automatically incremented whenever a Limb is created.

   d. It must contain a private, string-based field, called LimbDescription, for assigning a description to the Limb.

   e. The constructor should receive the string-based description of the Limb as its parameter.

   f. The class must contain a method, called CalculatePowerRequirements, which receives no parameters and returns a double value. This method should have no implementation of its own. Any derived classes should override this method.

   g. A Limb’s serial number should be created by using the current number of existing Limbs and combining it with the letter L. The serial number may never be less than 6 characters in length e.g. the very first Limb object should have a serial of L00001.
   h. An override of the ToString() method must be provided, which uses GetType() to return the type of the object, as well as its serial number, description and power requirements (rounded to three decimals). You may use your own discretion to format the returned string.

5. Develop a class called Arm. The class must be developed as follows:
   a. The class must inherit from class Limb.
   b. The class must have two private double properties. One for setting and returning the length of the UpperArm and another for setting and returning the length of the ForeArm. The properties must ensure that the UpperArm may only have lengths between 10 and 100 (cm) and the ForeArm may only have lengths between 20 and 80. If invalid values are received for the UpperArm or ForeArm, use your own discretion to provide default values.
   c. The constructor for the class must receive a description for the Arm, the length of its UpperArm and the length of its ForeArm.
   d. Class Arm should calculate its power requirements by multiplying the length of the UpperArm with the length of the ForeArm and determining the square root of the resulting value.
   e. An Arm’s serial number is calculated by taking the serial number created in the base class and adding the letter “A” to the end, e.g. L00001A.
   f. An override of the ToString() method must be provided, which returns all of the details provided by the base class’ ToString() method, but adds the length of the UpperArm and ForeArm.
6. Develop a class called Tentacle. The class must be developed as follows:
   a. The class must inherit from class Limb.
   b. The class must have a private double field, TentacleLength.
   c. The constructor for the class must receive a description for the Tentacle and its length.
   d. Class Tentacle should calculate its power requirements by dividing the length of the Tentacle by 4 and determining the square of the resulting value.
   e. A Tentacle’s serial number is calculated by taking the serial number created in the base class and adding the letter “T” to the end, e.g. L00001T.
   f. An override of the ToString() method must be provided, which returns all of the details provided by the base class’ ToString() method, but adds the length of the Tentacle.

Your program must make use of the main program shown below. Copy this code and place it in your own Main() method in the Program.cs file.

```
static void Main(string[] args)
{
    IBodyPart[] BodyParts = new IBodyPart[5];
```

```
        BodyParts[0] = new Eye("Brown");
        BodyParts[1] = new Head("Green", "Blue");
        BodyParts[2] = new Arm("Left arm", 40, 70);
        BodyParts[3] = new Arm("Right arm", 0, 200);
        BodyParts[4] = new Tentacle("A tentacle", 90);

        foreach(IBodyPart Current in BodyParts)
             Console.WriteLine(Current.ToString());

        Console.ReadLine();
    }
```

Your program should provide similar output to that shown below. Remember, yours may be formatted differently and the “Assignment1” text will be replaced by whichever name you gave your project.

```
Assignment1.Eye:  E00001, Brown
Assignment1.Head:  H00001, Assignment1.Eye:  E00002, Green, Assignment1.Eye:  E0
0003, Blue
Assignment1.Arm:  L00001A, Left arm, 52.915, 40, 70
Assignment1.Arm:  L00002A, Right arm, 50.000, 50, 50
Assignment1.Tentacle:  L00003T, A tentacle, 506.250, 90
```