Assignment 1: Console Application

Part I – Pet Owner

1. Objectives

- To create your first C# program as a Console Application with two classes
- To work with some built-in data types such as int, string and bool.
- To exercise with some useful methods of the **Console** class.
- To learn how to interact with the user to receive input and display output.

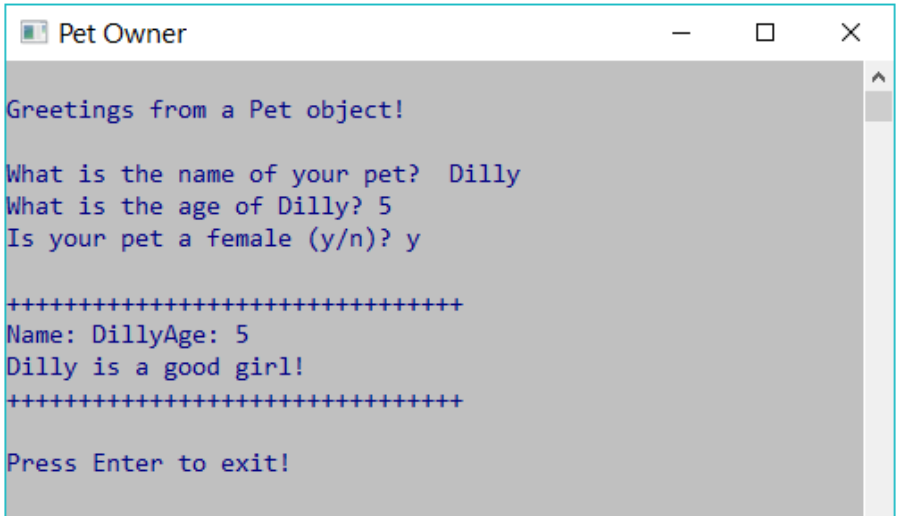
2. Description

This assignment consists of two parts, a mandatory part in which the highest grade will be a G (Pass), and a second part that is optional but required for the grade VG (Pass with Distinction). In the first part, you are given a problem with instructions to solve it. In the next part, you may find and implement a task similar to Part 1 by yourself.

3. Part 1 - Pet Owner

In this part, create a new Console Application project called Assignment1A. Create a sub-directory somewhere on your hard disk for this assignment and save your project there.

The application should ask the user for the name, age, and gender of a pet (or any animal). The values are to be saved in the program and then the program should display the information back to user. Here is a run example:



3.1 Requirements Part 1

3.1.1 Write a class with the name **Pet** and three fields (instance variables): **name** (string), **age** (int) and **isFemale** (bool). Save the class as **pet.cs** on your hard disk

3.1.2 All the three fields should be declared as private.

3.1.3 Write at least two methods in the **Pet** class, one for reading user input (the above three values) and one that displays the pet information as in the previous figure.

3.1.4 Write a start class "**PetOwnerMain**" and save it as **petownermain.cs** in the same directory as the **pet.cs** file.

3.1.5 Write a **Main** method, create an object of **Pet** and call the methods of the object to get an output as shown in the run example.

3.1.6 Control of user input is not mandatory in this assignment. Expect that users provide valid values.

If you need help and guidance, step-by-step instructions are provided in a separate document (Assignmetn1Help).

4. Part 2 - Program your own object

This part is only for grade G (Pass) and can be skipped if you are not going for a VG (Pass with distinction).

Look around at home, at your work, or wherever you are at this moment. You will find numerous objects, a chair, a baby, a TV, a house, or a car. Choose your **one** favorite object and program it as in Part 1.

4.1 For the object that you have chosen, include at least three fields that best describe the objects of this type. Determine then a data type (string, int, double etc.) that each attribute can be represented by.

For an **Album** class, the name of the singer, number of tracks, and release date could be some good fields. These can be declared as string, int and **DateTime.** Think about at least two operations (methods) that can be performed on the object using its fields.

The object does not have to be a physical thing; it may be a conceptual object, a calculation object, or an object like “Recipe”, “Movie”, or Address. For an **Address** class, street, city, and zip code are good attributes for most purposes. All of these three attributes can be declared as string, as they all contain a sequence of characters (even the zip code).

4.2 Create a new Project in the same Solution as Part I, or choose a new Solution in VS. Follow the same pattern as in Part 1 to solve the problem:

4.2.1 You must create two classes as in Part I.

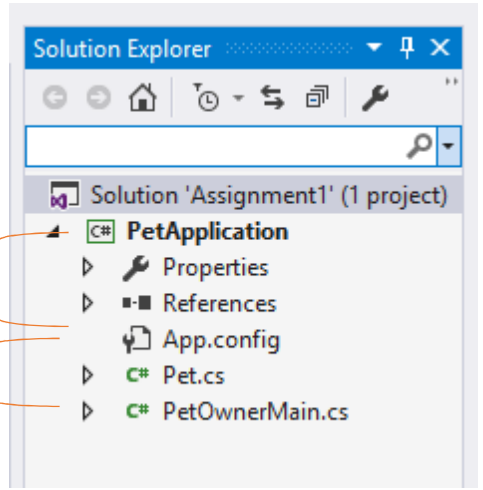
4.2.2 All instance variables must be declared as private. No variable should be declared public.

5. Help and Guidelines

5.1 The Project

Start Visual Studio and create a new Console Application. Visual Studio will prepare a frame for your project including a solution, a project with the required properties and references and a start-up cs-file (program.cs). Give your application, your project, and the first class (program.cs) appropriate names. The figure below shows an example that you can follow. The names **Assignment1**, **PetApplication**, **Pet** and **PetOwnerMain** are my choices.

To change the default names, just right-click on the name, select **Rename** from the context menu, and write a new name.



5.2 The Pet class

Start your coding by creating a new class. Right-click on the project name (**PetApplication**), select **Add**, and then **New Item** (or Class down the menu). If you go through the option **New Item**, select **Class** from the template, give a class name (Pet.cs), and click the button **Add**.

Declare the three variables:

```
class Pet
{
    private string name; //name of the pet
    private int age;     //age as an integer
    private bool isFemale;  //true if female,  false otherwise
```

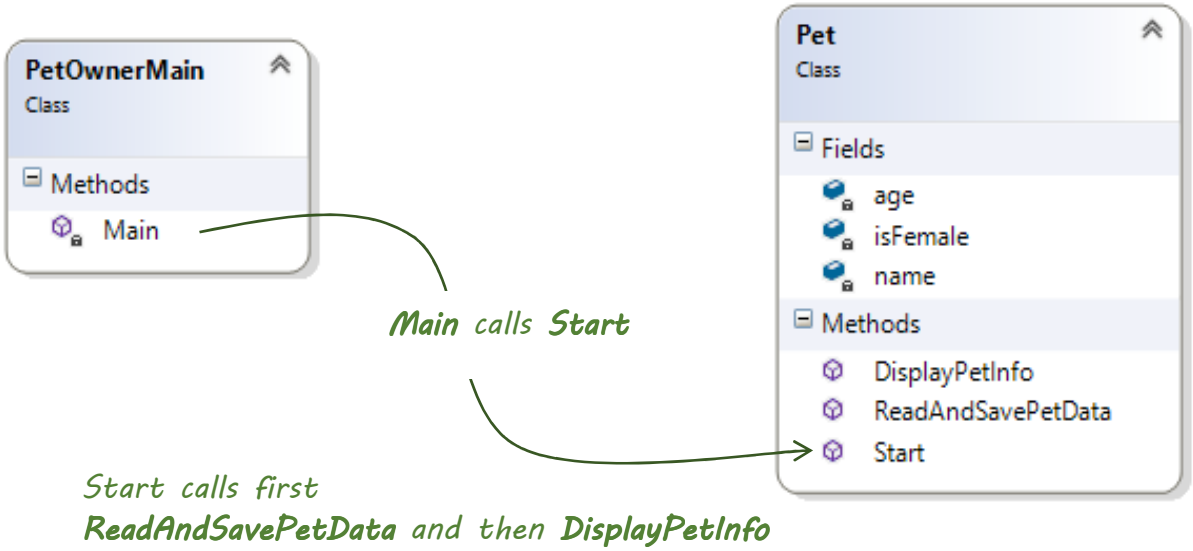
You can write the comments above the variable declarations too, but short comments can be written on the same line as in above image. Some prefer to write comments under a code line, but this is perhaps not very usual. What is important is that your programming style is consistent.

Now that you have variables to store the necessary data, the next step is to write methods. Methods are to do operations. In this class, we expect the objects of the class to perform two simple operations:

- Interact with the user and get values for the name, age and if the pet is female, and save the values in the variables of the object.
- Display the data saved in the object back to the user.

Write these two methods in the class **Pet**.
Then go to the class **PetOwnerMain** (or Program if you have not changed the name) and write code to create an object of the **Pet** class and call the above methods.

The following figure give you a clue on variables and methods of the two classes (Pet and PetOwnerMain) classes that you need to write in this project.



You may of course call the two methods **ReadAndSavePetData** and **DisplayPetInfo** from the Main method (as in this construction), but to keep the Main method shorter and to let Pet class control the flow of the code in a correct order, you can write a method in the Pet class that encapsulates the two calls. If we name this method Start, it may look like this:

```
public void Start ( )
{
    ReadAndSavePetData ( );
    DisplayPetInfo ( );
}
```

In order to give you a chance to think and try the solution by yourself, no more instruction is provided here. Instead, a step-by-step guidance is provided in a separate document. You can try solving the assignment by yourself but refer to the solution when you are unsure or need help.

6. Submission

Compress all your files include the Properties folder using a **zip** or **rar** format (as explained in the document “Quality Standards and Guidelines”). Go the Assignment page (where you downloaded this document) in Its L, click the Submit Answer button and upload the compressed file. Do not upload singles files and do not send your solution by email even when Its L is not down (doesn’t happen often). Its L is the only way to make your submission and it is there, your teachers can record your grade.

Good Luck!

Programming is fun. Never give up. Ask for help!

Farid Naisan,
Course Responsible and Instructor