MALMÖ HÖGSKOLA
Malmö University

Programming Using C#, Basic Course

Assignment 6a
OOP - Inheritance

Customer Registry Version 2

(G or C grade only, for higher grade do 6b instead!)

Mandatory

Farid Naisan
University Lecturer
Department of Computer Sciences
Malmö University, Sweden

Assignment 6- Inheritance

1. Objectives

In the last module, we practiced with the first of the three important aspects of OOP, encapsulation. Now it is time to move forward and try the next concept, inheritance. Object inheritance helps to reuse code by creating an "**is a"** relation between objects of same type. The main topics covered in this assignment are:

- to implement inheritance,
- to practice using a TabControl.

This is a small assignment that gives only a passing grade G(Swe), or C (ECT). If you wish to receive a higher grade, you must do Assignment 6b (Universal Calc).

‘

2. Description

You programmed a customer registry application in the previous assignment. The OOP model included mostly a "has a" relation as we had not covered the topic of inheritance at that time. In this assignment, we will examine the model to implement inheritance to optimize our solution. Meanwhile, we will define a customer more generally.

2.1 A Customer can be:
- 2.1.1 A private person, with a name, personal number, etc.
- 2.1.2 A company, with a name
- 2.1.3 A prospect (may become a client)
- 2.1.4 A client (has become a customer)

2.2 A customer has a contact and a contact has Address,Email and Phone.
2.3 The rest as in the last assignment.

3. Requirements:

3.1 You must implement inheritance at least once.¨ All class definitions, and class methods are to be documented using a XML-compatible format

3.2 All classes must contain at least one constructor and they must define properties.

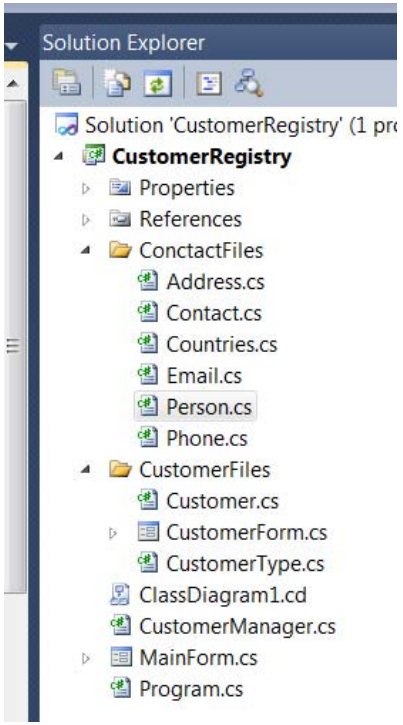
3.3 The program must work satisfactorily as in the last assignment.

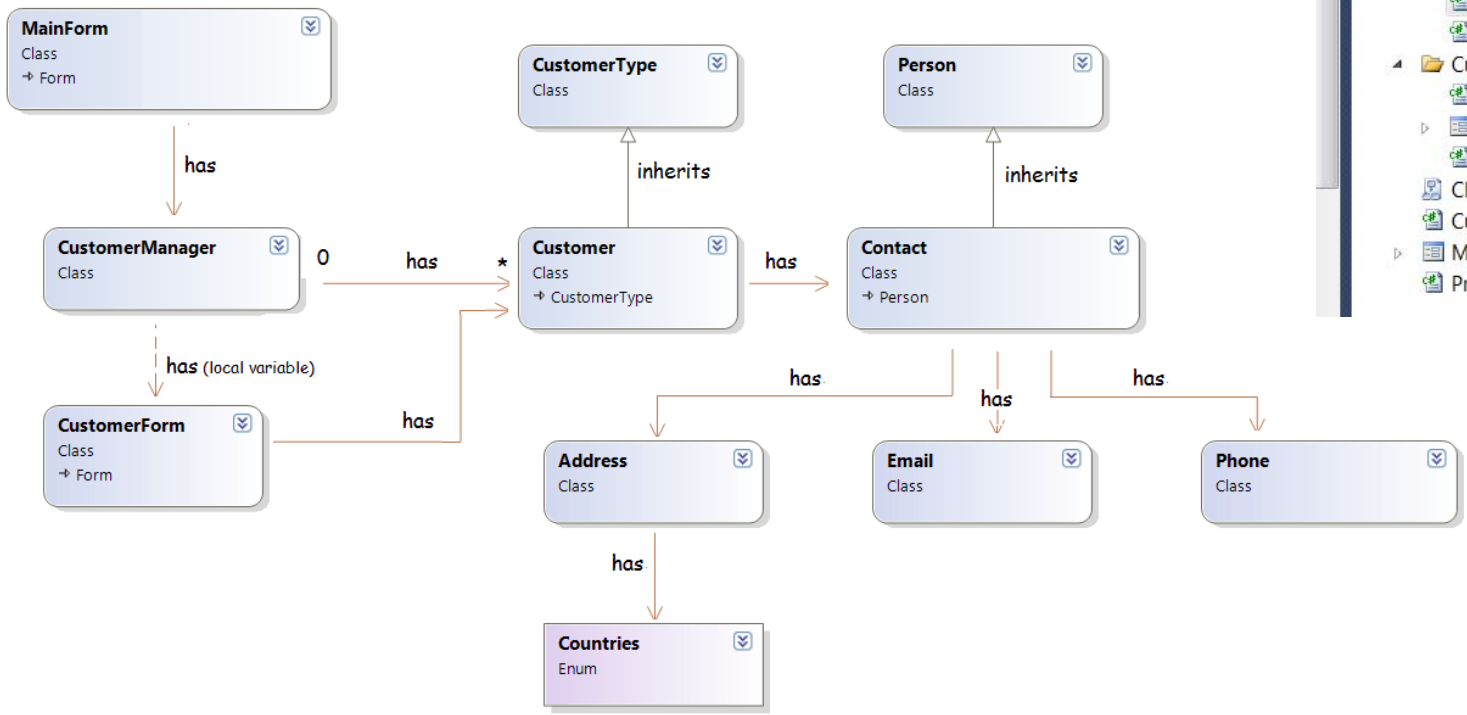
4. To Do

4.1 Having the above requirements in mind, try to remodel the last assignment and then implement the changes in your project

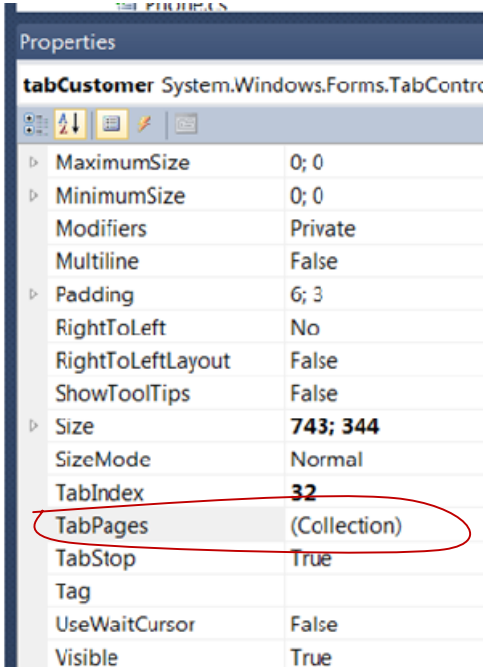
4.2 Take a good time to understand the following class diagram and in case you come up with better ideas to implement inheritance, don't hesitate to discuss it on the forum, and if you are sure about it, implement it in your code..

4.3 When you are certain that you have a clear idea of how the classes and relations between them, you can begin coding. Copy your Customer Registry project to a new folder..Implement changes in your code, compile and test before submission.





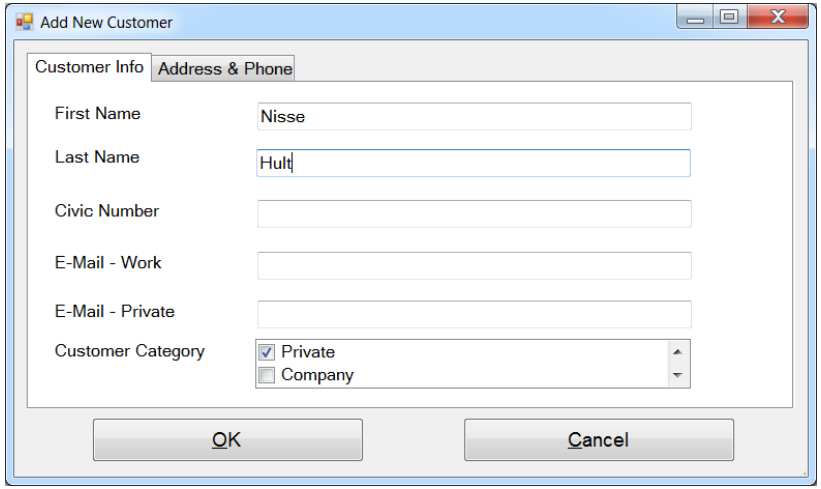
4.4 Because of the amount of information on the CustomerForm, use a TabControl to organize the input into pages. Draw a TabControl from the ToolBox in Visual Studio on the Form and then change the Properties of the TabControl using the Property Windows. To change the properties of the tabs (pages), click on the "Collection" value of the TabPages as marked on the figure. Then you can work with designing and coding of controls as you do when working with a form.



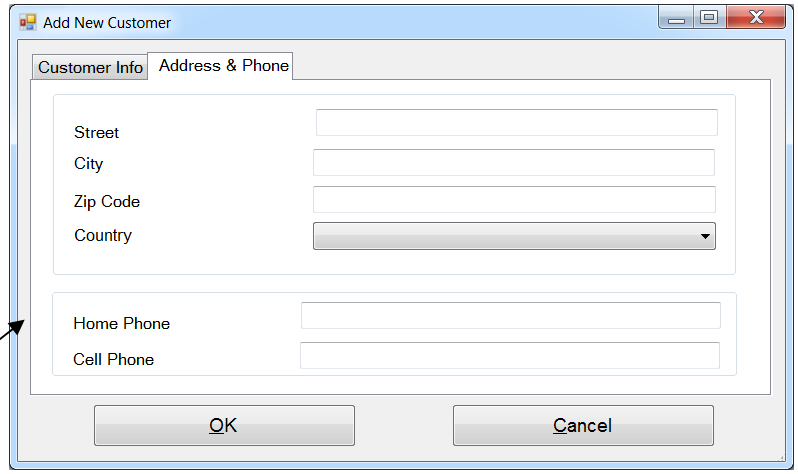
4.5 In the figure below a CheckedListBox is used to allow the customer types to select multiple options, but this is not mandatory. You can use a group of checkboxes or another control that you like to practice with. If you choose to use a CheckedListBox, you can use the Property SetItemCheck to put a checkmark on an entry (1 is the row number in the list, true = checked).:

```
lstCustomerType.SetItemChecked(1, true);
```

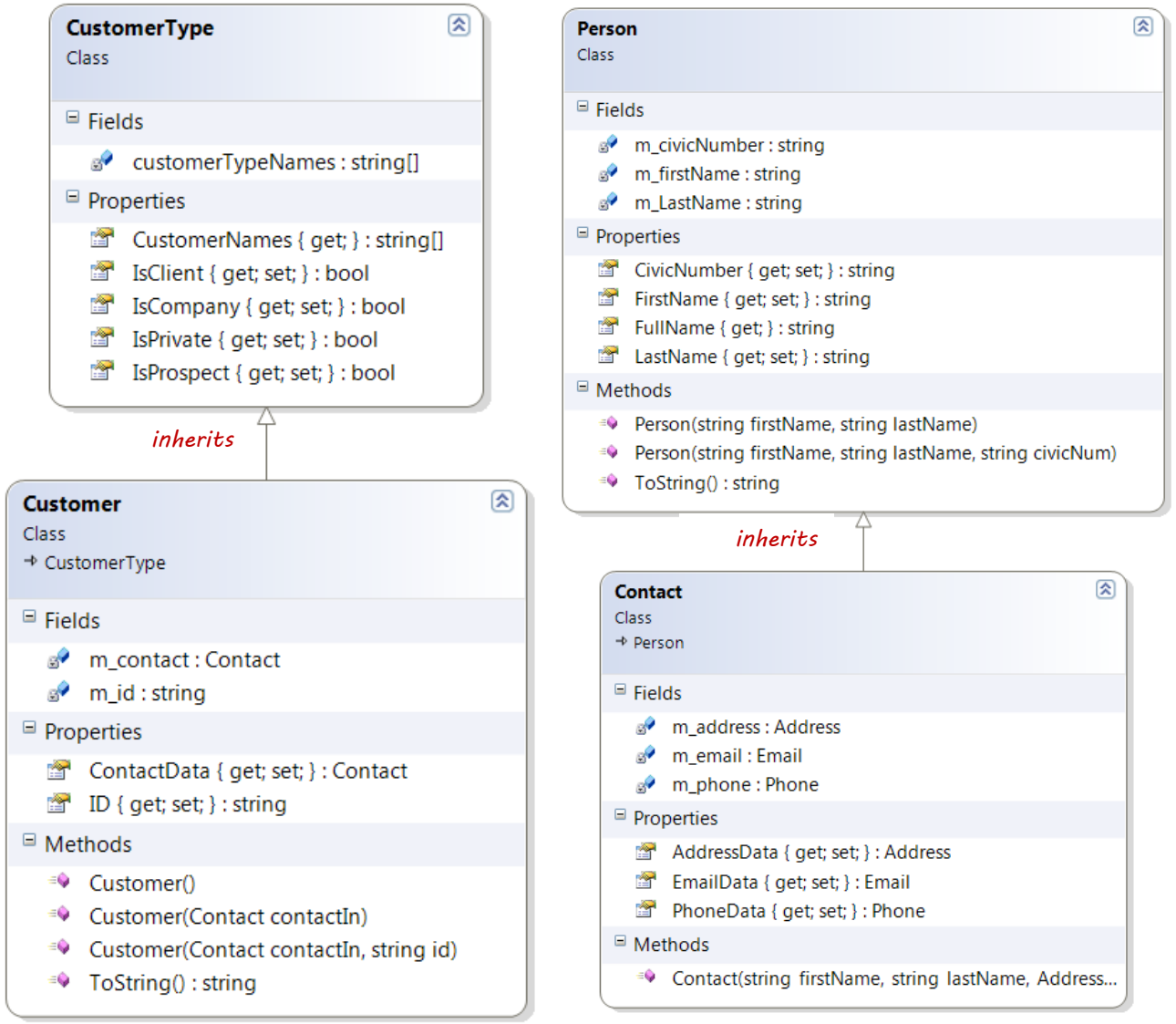
4.6 You can design the pages of the TabControl as you wish. Below is an example.



4.7 When the second tab is clicked, the view changes as shown here:



The classes affected are shown here.



Good Luck!

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

Farid Naisan
Course Responsible and Instructor