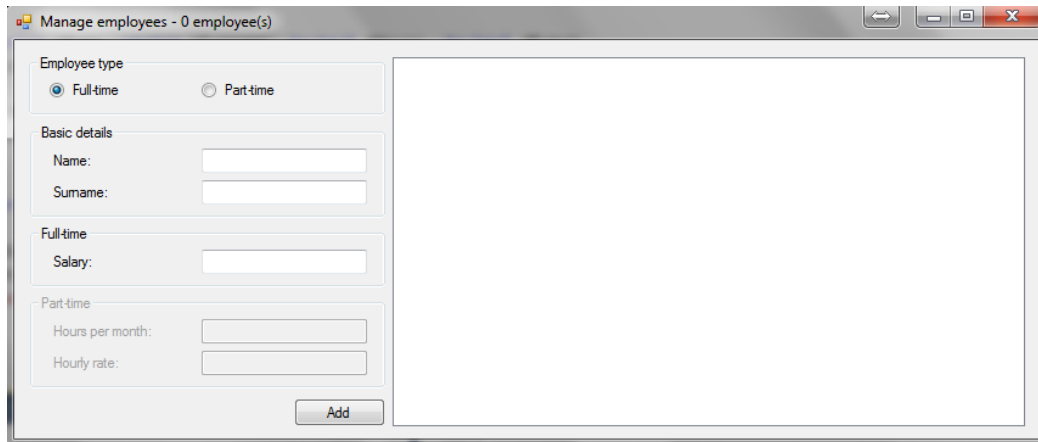


MSE1122 Assignment 2 2014

BizCo has contacted you to develop an employee management system for them. Your sales consultant has provided you with the following specifications:

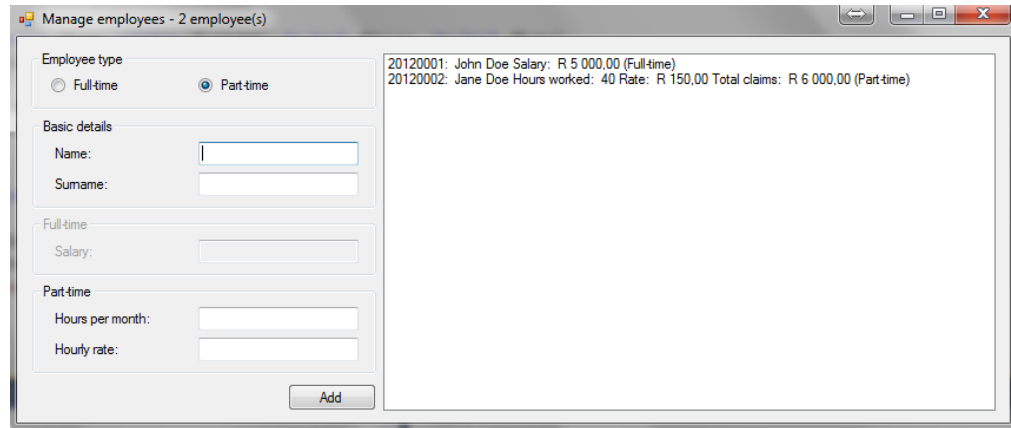
1. Create a class named Employee with the following details
 - a. Three private string fields to store an employee number, name and surname.
 - b. A private static counter to keep track of the total number of employees created.
 - c. A constructor which
 - i. Receives an employee's name and surname.
 - ii. The employee number should be constructed by using the current year (using a DateTime object) and the current employee count padded with leading zeroes, so that the employee number is always 8 characters long, e.g. If the tenth employee was created in 2013, the employee number would be 20130010. You do not have to reset the employee counter every year.
 - iii. Initializes and increments fields as/if necessary.
 - d. An override of the ToString() method which returns a string similar to: [EmployeeNumber]: [Name] [Surname]
 - e. Make sure it is possible to call Array.Sort for an array of Employee objects. The sorting must be done according to the Employee surnames.
2. Create a class named FullTime, which inherits from class Employee, with the following details:
 - a. A private decimal field to store salary.
 - b. A constructor which
 - i. Receives an employee's name, surname and salary.
 - ii. Initializes and increments fields as/if necessary.
 - c. An override of the ToString() method which returns a string similar to: [EmployeeNumber]: [Name] [Surname] Salary: [Salary (as Currency)] (Full-time)
3. Create a class named PartTime, which inherits from class Employee, with the following details:
 - a. Two private decimal fields to store hours worked and hourly rate.
 - b. A constructor which
 - i. Receives an employee's name, surname, hours worked and hourly rate.
 - ii. Initializes and increments fields as/if necessary.
 - c. An override of the ToString() method which returns a string similar to: [EmployeeNumber]: [Name] [Surname] Hours worked: [HoursWorked] Rate: [HourlyRate (as Currency)] Total claims: [Calculation (as Currency)] (Part-time)

4. In order to keep track of all the Employees, create a GUI similar to the following:



5. Basic details (name and surname) must be entered for every employee.
6. When the full-time employee option is selected, the part-time employee entry fields must be disabled. When the part-time employee option is selected, the full-time employee entry fields must be disabled.
7. Ensure that it is not possible to add an employee if all the necessary fields have not been completed.
8. Make use of exception handling to ensure that the program does not “bomb out” if something goes wrong, e.g. a non-decimal value is entered into the salary, hours or rate fields.
9. When the user clicks on the Add button the following must happen.
 - a. The error checking mentioned in points 7 and 8 must be applied.
 - b. A new FullTime or PartTime object must be created if the conditions in 7 and 8 are met.
 - c. This newly created object must be added to a global array containing both the FullTime and PartTime objects (i.e. not 2 separate arrays). The array must be capped at 10 objects. If the array is full, disable all functionality on the form.
 - d. Clear all the text fields and set focus back to name.
 - e. Call another method (you may choose a name) which does the following:
 - i. Clear the listbox of all entries.
 - ii. Sort the array of FullTime and PartTime employees using Array.Sort.
 - iii. Use a single loop to process all the FullTime and PartTime employees in the array and then add their details to the listbox.
 - iv. Update the form’s heading to include the total number of employees.

- f. The screenshot below is an example of what your screen should look like after adding 2 employees.



The screenshot shows a window titled "Manage employees - 2 employee(s)". On the left, there are input fields for adding a new employee. The "Employee type" section has two radio buttons: "Full-time" and "Part-time", with "Part-time" selected. Below this, the "Basic details" section has "Name:" and "Surname:" labels followed by text input boxes. The "Full-time" section has a "Salary:" label and a text input box. The "Part-time" section has "Hours per month:" and "Hourly rate:" labels, each followed by a text input box. An "Add" button is at the bottom of the input section. On the right, a list of employees is displayed:

- 20120001: John Doe Salary: R 5 000,00 (Full-time)
- 20120002: Jane Doe Hours worked: 40 Rate: R 150,00 Total claims: R 6 000,00 (Part-time)

Submission Details

- You are allowed to work in groups of 3 students.
- Add all your student numbers and names to the top of all your **.cs** files as comments.
- Zip your entire project folder, rename it to match one of your project members' student number and submit via the subject's Moodle site before 09:15 on 16 October 2014.
- If an assignment has been handed in and your name or student number is not contained in the document comments or in the file name, then you were not a part of the group. Don't ask afterwards to be added to it.