TEST APPLICATION’S DESCRYPTION

This application represents a simple functionality to view foo company’s staff for any date.

The main look of this application previewed on below picture:



**USER GUIDE**

By default, application displays all staff of the company setting “Date” field to current date. Staff representing by table, which consists of following columns:

* Name – The name of employee. The name has indent according to employee’s level in common hierarchy.
* Position – The Employee’s position at the company. In sample version they are: Employee, Manager and Seller
* Hire date – Employee’s date of start
* Base salary – Employee’s salary when he\she was hired
* Salary – Employee’s salary calculated on specified date

Last row of the table shows total value of whole staff salary – the sum of all employee’s salaries.

All rows also sorted according to employee’s subordination level and subordinary employees always placed lower than their boss.

User also can view staff on any date changing the “Date” field and clicking on “Show Staff” button.

**TECHNICAL GUIDE**

All members of application’s main classes are described in the MainClassesMap.dgml and MainClassesMap.jpeg files. The entire structure of the application is based on two main classes "Employee" and "EmployeeType".

The "Employee" class contains the "Boss" property, which points to its boss. This field is used to create a hierarchical structure of staff. The Type property points to an instance of the “EmployeeType” class. To optimize salary calculations, the calcDate and calcSalary fields are provided. The "Staff" property is a static list of all the company's employees and is filled in when a new employee is created.

Main methods:

public virtual (double Salary, double Total) CalcSalaryForDate(DateTime date) - a method designed to calculate the salary of an employee and the summary value of salaries of subordinate employees on a specified date. If a calculation of current employee’s salary has already been made on the specified date, the salary value is taken from the calcSalary field.

protected virtual double GetEmployeesSalaries(DateTime date) - a method designed to calculate the total salary of subordinate employees. Returns 0, because by the task’s condition an ordinary employee cannot have subordinates.

The "Manager" and "Seller" classes are inherited from the "Employee" class. Since the algorithm for calculating the salary of a manager and a seller differs only in calculating the total salary of subordinate employees, the GetEmployeesSalaries method is overridden in these classes. In the “Manager” class, the GetEmployeesSalaries method returns the sum of the salaries of only the first-level employees, and in the “Seller” class, the sum of the salaries of all subordinates.

The "Company" class is connected as a service and provides the FillStaff method for initializing company’s stuff and the property IEnumerable<Employee> Staff.

In addition to the task, the employees are ordered and employee’s name indented according to the hierarchy. This makes it easier for visual experience.

The current solution uses recursion to calculate employee salaries and this can affect performance with a large number of employees and deep nesting. In order to use this application for real purposes, it is needed to add functionality for creating, editing, and deleting instances of the "Employee" and "EmployeeType" classes with all the necessary checks. Adding the calcDate and calcSalary fields solves the performance problem, but complicates the Create, Edit, Delete methods of the “Employee” class.