ReadMe

Сопроводительный документ сделан в формате описания выполнения тестового задания.

Тестовое задание

Для отдела HR одной компании нужно написать приложение расчета заработной платы.

В компании работают сотрудники, характеризующиеся именем, датой поступления на работу, группой и базовой ставкой заработной платы.

Есть 3 группы сотрудников - Employee, Manager и Salesman. У каждого сотрудника может быть начальник. У каждого сотрудника кроме Employee могут быть подчинённые.

Зарплата сотрудника рассчитывается следующим образом:

* Employee - это базовая ставка плюс 3% за каждый год работы, но не больше 30% суммарной надбавки.
* Manager - это базовая ставка плюс 5% за каждый год работы, но не больше 40% суммарной надбавки за стаж работы. Плюс 0,5% зарплаты всех подчинённых первого уровня.
* Salesman - это базовая ставка плюс 1% за каждый год работы в компании, но не больше 35% суммарной надбавки за стаж работы. Плюс 0,3% зарплаты всех подчинённых всех уровней.
* У сотрудников (кроме Employee) может быть любое количество подчинённых любой группы.

**Требуется:** составить структуру классов, описывающих данную модель, а также реализовать алгоритм расчета зарплаты каждого сотрудника на произвольный момент времени, а также подсчёт суммарной зарплаты всех сотрудников фирмы в целом.

**Замечание:** при реализации тестового задания необходимо предположить, что вы разрабатываете не просто прототип, а систему enterprise уровня, соответственно важнее продемонстрировать архитектурно более красивое решение, даже в ущерб быстродействию. Не обязательно реализовывать всю архитектуру в полном объёме, не реализованные или упрощённые моменты нужно прокомментировать.

**Решение:** нужно сделать на C# с использованием sqlite, применяя любые библиотеки. Код передать в виде репозитория git на github или аналоге.

Весь функционал, запрашиваемый в тестовом задании, реализован в полном объёме. Архитектура реализована через отдельные функции запроса данных в SQlite. После этого, результаты запроса обрабатываются через типизированный список List, состоящий из массивов строк. Далее определяется уровень доступа для нынешнего пользователя и вычисляется, в зависимости от доступа, зарплата для работника или работника и его подчиненных. Входные данные для всех запросов реализованы через стандартные элементы Windows Forms. Сама база данных разбита на три таблицы, описывающие работников, отдел(несколько менеджеров под одним salesman) и должности сотрудников. Доступ к созданию/удалению данных в базе осуществляется посредством доступа суперадминистратора базы. Упрощено его нахождение внутри базы, вынесен условием проверки на функции кнопок.

Список логинов и паролей тестовой БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | login | password | position | Comments |
| 0 | me | 281295 | superadmin | Вне таблиц |
| 1 | ASU | 1111 | Salesman |  |
| 2 | ASUman1 | 2222 | Manager |  |
| 3 | ASUman2 | 3333 | Manager |  |
| 4 | ASU1 | 4444 | Employee |  |
| 5 | ASU2 | 5555 | Employee |  |
| 6 | ASU3 | 6666 | Employee |  |
| 7 | ITsales | 7777 | Salesman |  |
| 8 | ITman | 8888 | Manager |  |
| 9 | IT1 | 9999 | Employee |  |
| 10 | IT2 | 10101010 | Employee |  |

Не реализовано:

-защита от сторонних данных. Пожертвовал временем на них в угоду функционалу.

-не реализованы тесты.

**Дополнительные плюсы:**

* Написан краткий обзор решения тестовой задачи, описана архитектура, ее плюсы и минусы (что можно улучшить, поменять или еще какие-то соображения для использования решения в реальных целях) +
* Код покрыт тестами.
* Программа имеет графический интерфейс. +
* Будет возможность просмотреть для выбранного сотрудника список его подчинённых. +
* Будет возможность добавлять новых сотрудников разных видов +
* Будет возможность разграничения прав, каждый сотрудник будет иметь свой логин/пароль, имея возможность просматривать только свою зарплату, и зарплату своих подчинённых. Также должен быть супер-пользователь, который имеет доступ ко всем. +