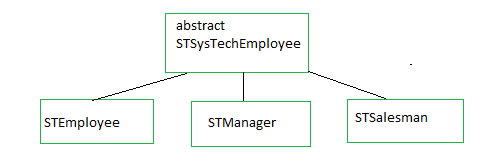
# Обзор решения.

Объектная модель –



Абстрактный класс **STSysTechEmployee**, в котором описываются общие для всех сотрудников поля:

**Name**,

**BaseRate**(базовая ставка),

**parentId** (ссылка на начальника) ...;

абстрактный метод **GetRate** (расчет зарплаты), который должен быть реализован для всех групп.

От него наследуются **STEmployee**, **STManager**, **STSalesMan**. В каждом из этих классов **GetRate** определен с соответствующим алгоритмом.

Для получения данных используется статический класс **STSysTechEmployeesData** со списком **List<STSysTechEmployee>** - при запуске приложения, данные из базы загружаются в этот список и программа работает с ним. При добавлении сотрудника, данные обновляются.

Плюсы такой архитектуры:

1. Для каждой группы можно добавить в класс свои свойства, легко найти и поменять алгоритм **GetRate.**
2. Не надо городить if\else везде чтобы определить – это менеджер или нет. Достаточно на этапе выгрузки данных из бд получить их тремя группами и добавить в общий список объекты типа **STEmployee, STManager, STSalesMan.**
3. Если потребуется дополнительно вычислять премии, штрафы, любые другие величины- эти методы легко добавить в каждый класс. На этапе выгрузки данных ничего менять не придется.

Минусы:

1. Загрузка сразу всеx данных в статическое поле класса занимает память, нужно делать конкретные запросы под необходимую операцию, чтобы использовать память под минимум необходимых данных.
2. Хранение в бд с помощью parent\_id – не лучшая идея. Лучше использовать **PostgreSql**, в котором есть модуль **ltree**.
3. **Entity framework** тоже лучше использовать для данной задачи, а не запросы в строку.

Графический интерфейс

Возможные улучшения:

1. Валидаторы на ввод.
2. Главная лента действий вверху(добавление сотрудника\расчет за –и еще что понадобится).
3. Иконки вместо кнопок – они компактнее и привычнее современному пользователю.
4. Добавление сотрудника – выбор начальника можно сделать формой с treeview контролом и там уже выбирать сотрудника.

