**Логика программы лабораторной работы 4**

Первое что делаем, это генерируем справочник. Для этого создаем имена, коды телефонов и сами телефоны. Количество абонентов указывается в константе в самом начале класса.

Как показали эксперименты, если номера телефонов сохранять в массиве строк, генерация происходит значительно дольше, чем числовые типы.

Тип int брать не получится, т.к. номер телефона это больше чем вмещает int, поэтому выбран long. С ним генерация происходит быстрее.

После генерации идет самый трудоемкий процесс, а именно заходить к каждому абоненту и смотреть каждый номер из контактов. Таким образом считается статус абонента

Самых же активных абонентов программа выводит в конце.

Однако временные затраты равны n2, а значит, чем больше абонентов, тем дольше будет прочем поиска абонентов и тем вероятнее падение программы из памяти.

При отключенном методе поиска абонентов, программа способна сгенерировать 1 000 000 абонентов, достаточно быстро.

Если включить вывод всех абонентов, это так же нагружает память и при небольших мощностях, программа может упасть.

Однако до 900 000 абонентов при 4гб оперативной памяти, программа справляется быстро.

Если отключить общий вывод и включить поиск, то нормально программа может справится при 100 000 абонентов.

**Вторая версия программы**

Во второй версии немного изменена логика работы программы в отличие от первой версии

А именно основная коллекция теперь TreeMap, как и коллекция номеров

Так же цикл поиска номеров оптимизирован за счет break

Так же для поиска был использован метод contains

100 000 записей, обрабатываются за несколько минут