Имеется сторонняя система логирования, со своими плюсами и минусами. Она нам досталась в наследство и нам придется с ней работать, исправить её недостатки сейчас нет возможности. Требуется написать решение, которое будет обрабатывать сообщения из лог-файла, чтобы справиться с недостатками существующей системы логирования.

О сторонней системе известно следующее:

1. Сообщение имеет следующий формат:

```
dd.MM.yyyy HH:mm:ss.SSS <msg str1>
<msg str2>
<msg str…>
<msg strN>
CRC_<CRC>
```

где,

`dd.MM.yyyy HH:mm:ss.SSS`

`dd` - день месяца, например 01 - первый день месяца
`MM` - номер месяца в году, например Март обозначается как 03
`yyyy` - год, в формате 4х цифр
`HH` - час в сутках, в формате 24h
`mm` - минуты
`ss` - секунды
`SSS` – миллисекунды

`<msg str1> … <msg strN>`

строка сообщения, ограниченная только следующим
- строки сообщений не могут начинаться с `CRC_`
- строки сообщений не могут начинаться с даты формата `dd.MM.yyyy HH:mm:ss.SSS`
- длина каждой строки не превышает 1024 символа
- в сообщениях нет пустых строк

`CRC_<CRC>`

значение контрольной суммы сообщения, причем
- контрольная сумма считается алгоритмом MD5
- символы перевода строки и возврата каретки не входят в контрольную сумму
- контрольная сумма считается от всего сообщения (дата и строки сообщений)

2. Очередность строк сообщения всегда корректна

3. Сообщения могут накладываться друг на друга, в результате строки одного сообщения могут быть спутаны со строками другого, например:

   сообщение 1 строка 1
   сообщение 1 строка 2
   сообщение 2 строка 1
   сообщение 2 строка 2
   сообщение 1 строка 3
   сообщение 2 строка 3

4. Сообщения в лог файл попадают не в хронологическом порядке. Но задержка между отправкой сообщения на логирование и фактической записью не более 10 сек. В течение этих 10 секунд гарантируется, что любое другое сочетание строк не приведёт к той же контрольной сумме.

Что требуется:

1. Правильно “склеить” сообщения, при этом известно, что существует только одно решение
2. Вывести в хронологическом порядке. Если время 2х или более сообщений совпадает, тогда выводить их нужно в порядке появления в лог файле.

Требования к решению:

1. Оформить в виде maven проекта (заготовка проекта с интерфейсом и примерами прилагается)[1]
2. Java 1.7
3. Реализовать интерфейс LogParser
4. Класс решения должен быть в пакете “com.aci.jd2015”
5. Выделенная память - 64 Mb
6. Кодировка UTF-8

Критерий	**Способ проверки**	**Максимальное количество баллов**
Прохождение тестов на корректность решения	Автоматизированные тесты	0
Покрытие кода Unit-тестами и качество их написания	Code-review	10
Прохождение тестов на граничные условия	Автоматизированные тесты	10
Качество кода (соответствие Google Java Style, сложность методов, следование принципам ООП)	Code-review	10
Скорость выполнения	Автоматизированные тесты	10

Пример**:**

Входное сообщение

```
28.03.2015 01:31:27.025 И стоит береза
В сонной тишине,
28.03.2015 01:31:25.025 Белая береза
Под моим окном
Принакрылась снегом,
Точно серебром.
CRC_59427ae1242319d1092f6963fd481878
И горят снежинки
28.03.2015 01:31:28.025 А заря, лениво
В золотом огне.
CRC_20d261ebef3bd9f9a42c81612e9efa51
Обходя кругом,
обсыпает ветки
```

[1] Если вы не знакомы с Maven, прочитайте следующую статью http://habrahabr.ru/post/77382/

```
28.03.2015 01:31:26.025 На пушистых ветках
Новым серебром.
Снежною каймой
Распустились кисти
CRC_1633f049ff1a52ea31ac959ce9222486
Белой бахромой.
CRC_599692d99d7f399e35ad029212f74272
```

Выходное сообщение

```
28.03.2015 01:31:25.025 Белая береза
Под моим окном
Принакрылась снегом,
Точно серебром.
CRC_59427ae1242319d1092f6963fd481878
28.03.2015 01:31:26.025 На пушистых ветках
Снежною каймой
Распустились кисти
Белой бахромой.
CRC_599692d99d7f399e35ad029212f74272
28.03.2015 01:31:27.025 И стоит береза
В сонной тишине,
И горят снежинки
В золотом огне.
CRC_20d261ebef3bd9f9a42c81612e9efa51
28.03.2015 01:31:28.025 А заря, лениво
Обходя кругом,
обсыпает ветки
Новым серебром.
CRC_1633f049ff1a52ea31ac959ce9222486
```