# Javaトラブルシューティング

## Javaの問題

　java実装中、テスト中、本番稼働中と様々なところで問題が発生するが、その中でも特に解決が難しい以下の２つについてツールを使った問題分析と解決方法を確認する。

・メモリリーク

・デッドロック

## Java Mission Control(JMC)

本番稼働時のプロファイリングおよび診断を行うツール・スイート。

Java SE Development Kit 7, Update 40 （JDK 7u40）より、オラクルの Java HotSpot VM にバンドルされた。

元は、米BEAの開発したJrockit　Mission Controlだったものが、オラクルが米BEAを買収したことにより、統合された。

Java Management Extensions（JMX）コンソールと Java Flight Recorder(JFR)という２つのツールが含まれている。

レコーディング（プロファイリング）の仕組み自体、OracleのHotSpot JVMの中に入っているので、レコーディングによるオーバヘッドが少ない。

## Java Management Extensions（JMX）

GUI及び簡単なリアルタイムモニタリングが出来るツール。

　ライセンス不要で使える。

## Java Flight Recorder(JFR)

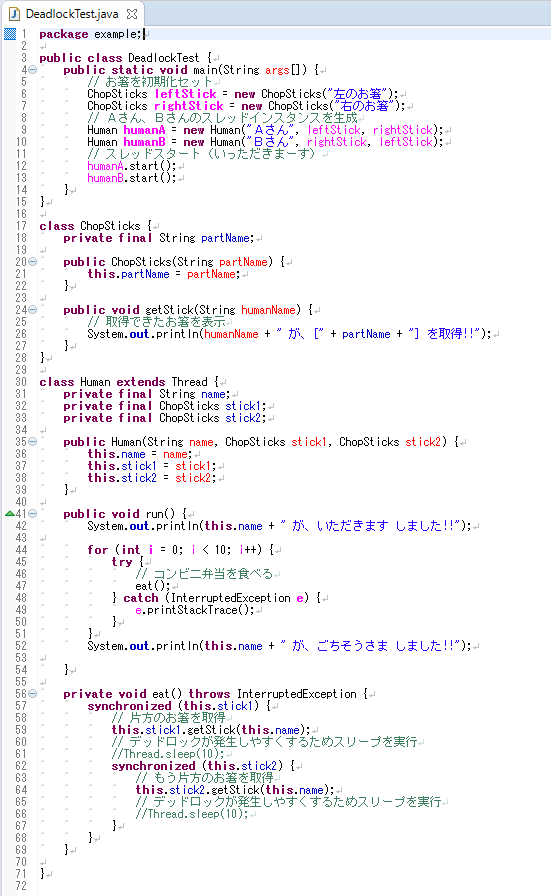
　Javaアプリケーションの動作をレコーディング（飛行機のブラックボックス - Flight Recorderにちなんだ製品名）するプロファイラー

記録は「.jfr」拡張子のファイルとして出力される

開発用途での利用は無料

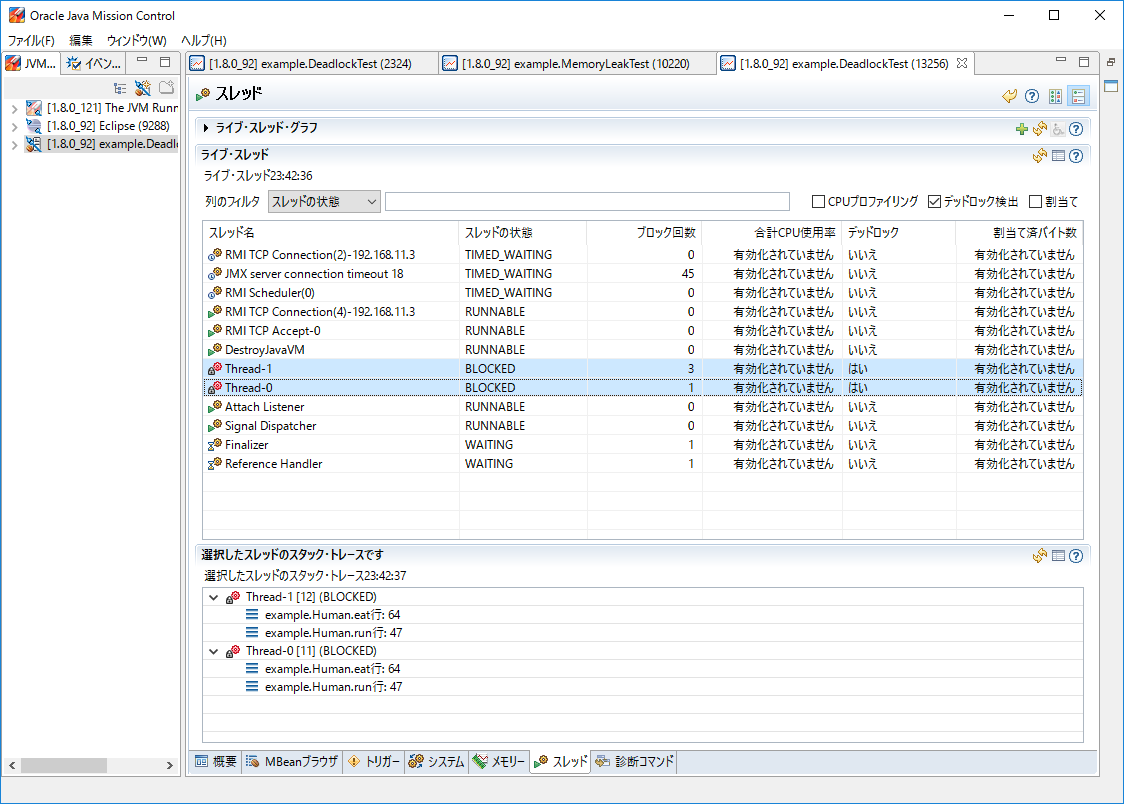
## JMXによるデッドロックの検出

サンプルコード



JMXによる確認

「スレッド」タブのデッドロッ検出をチェック



デッドロックしているスレッドが検出される。

デッドロック状況のスタック・トレースも表示可能

JFRによるメモリリークの調査

サンプルコード



実行引数を設定して実行

**-XX:+UnlockCommercialFeatures -XX:+FlightRecorder** -XX:StartFlightRecording=duration=60s,name=myrecording,filename=myrecording.jfr,settings="C:\\Program Files\\Java\\jdk1.8.0\_121\\jre\\lib\\jfr\myprofile.jfc"

duration=60sで指定した60秒間の時間限定レコーディングを行う。

duration指定がないと連続レコーディングを行う。

name= Java Flight Recorderの識別子を指定。

filename=レコード結果を記録するファイルを指定。

settings=profileで設定種類（設定ファイル）を指定。

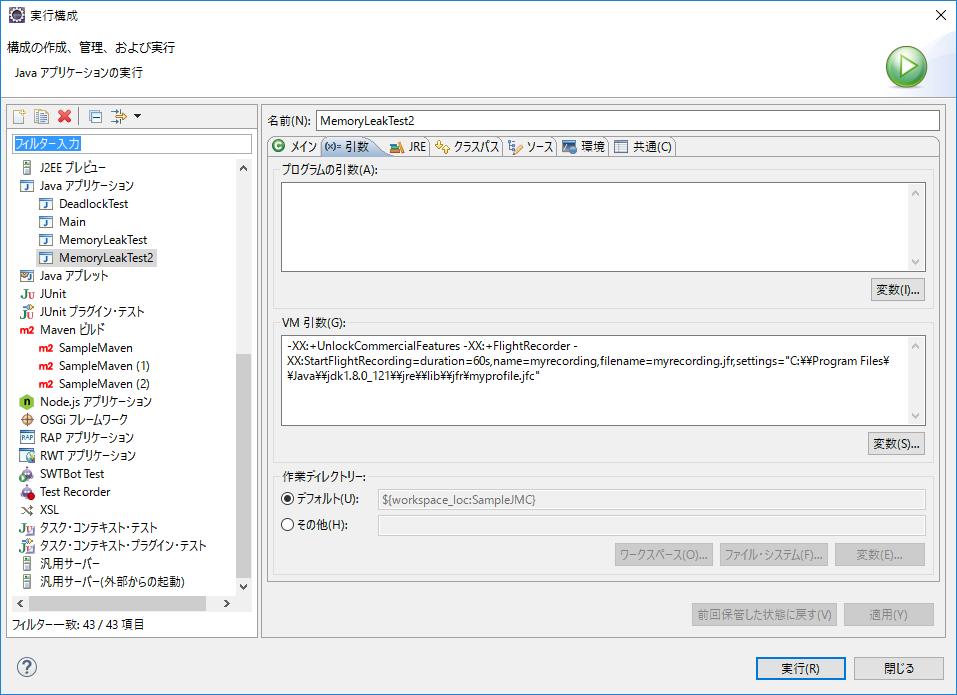
デフォルトの設定ファイルは

<JREディレクトリ>\lib\jfr\にある（default.jfc、profile.jfc）。

デフォルトのプロファイリングの設定だとメモリの割り当て状況(Memory -> Allocations)やヒープのオブジェクト解析(Memory -> Object Statistics)がオフになっているので、有効にする。

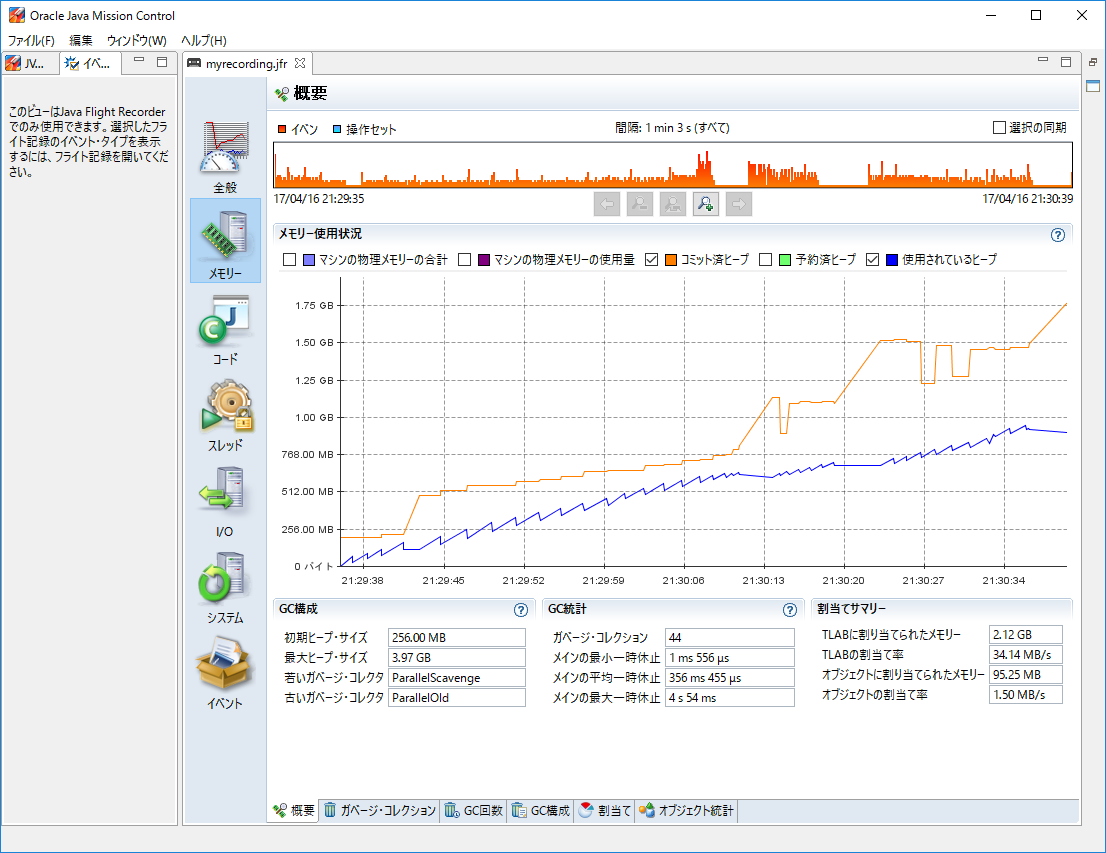
https://docs.oracle.com/javase/jp/8/docs/technotes/tools/unix/java.html

<http://waysaku.hatenablog.com/entry/2014/07/22/005304>



myrecording.jfrを開く

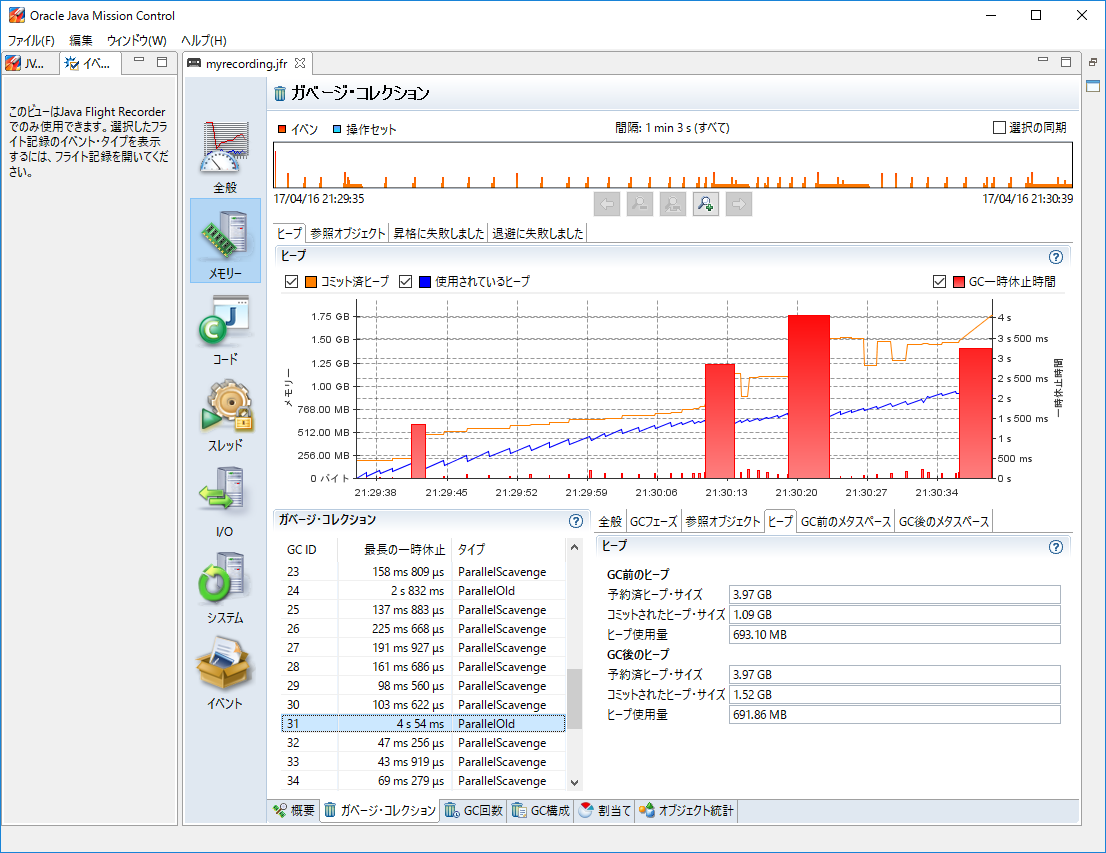
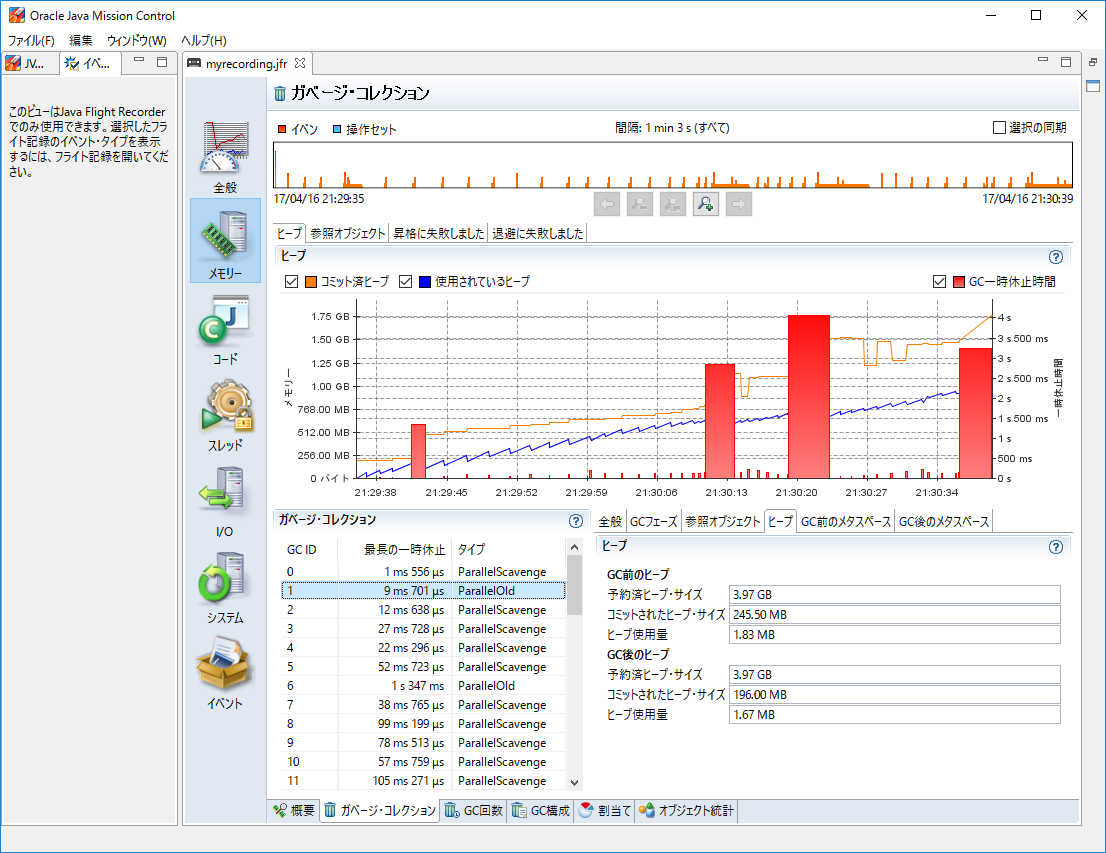
メモリーの概要でヒープの状態を確認



緩やかにヒープの使用量が増加していくのは、メモリリークの症状

メモリー・リークの検出

「メモリー」タブをクリックし、「ガベージ・コレクション」タブを選択して、最初と最後のOldコレクションを確認。



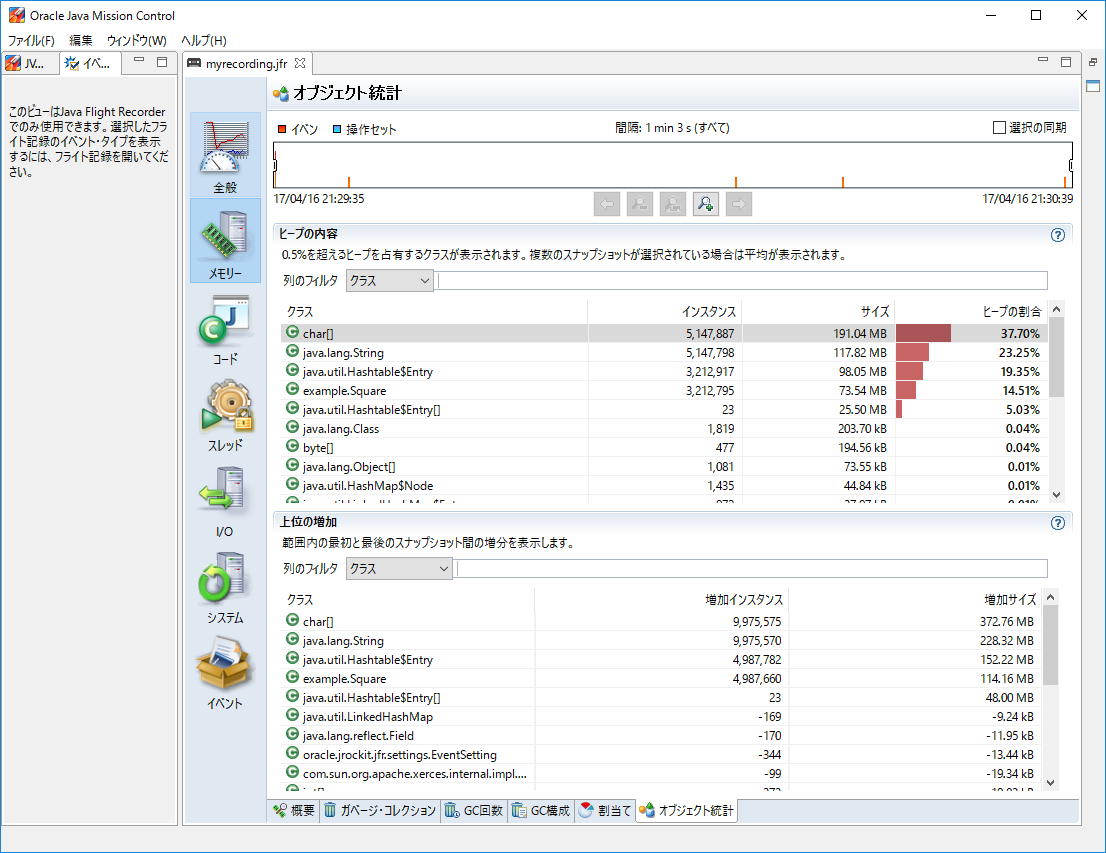
33回目のGCでは、約 1.24Mが解放

されるが、ほとんどのメモリが解放されていない

1回目のGCでは、約 0.16Mが解放

リーク・クラスの調査

「オブジェクト統計」を調べます。1つの長い記録を検討し、記録を通してヒープ使用量が最も増加したクラスを調べます。

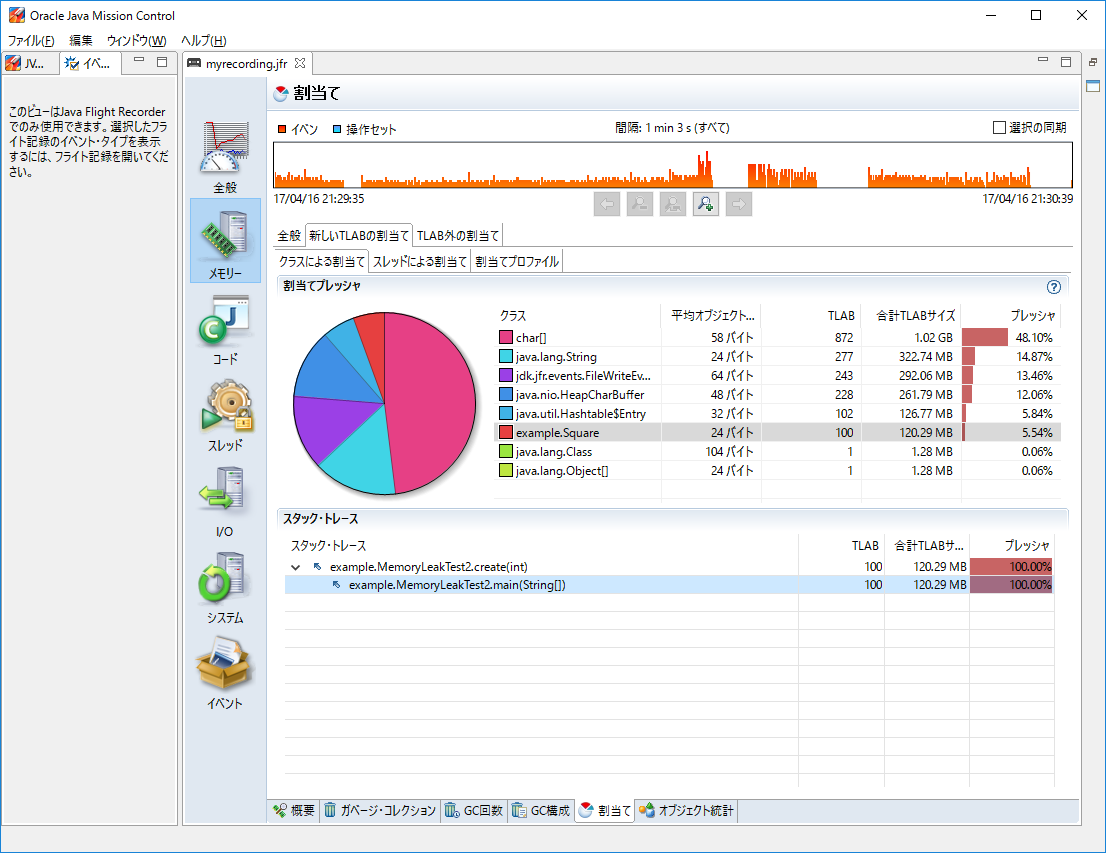


標準ライブラリの一部ではないクラスに注目。

多くの場合、多くのStringsが割り当てられているChar配列が表示される。これらのStringsを維持しているオブジェクトに注意する。

また、オブジェクトのサイズではなく、インスタンスの数でソートする。

特定のクラスのリークが予想される場合は、「割当て」の「新しいTLABの割当て」タブを確認する。



Squareクラスのメモリ割り当て時のスタック・トレースなどが確認できる。

※TLAB・・・ (Thread Local Allocation Buffer) Java ヒープ中でのメモリ確保処理 (= Java オブジェクトの生成処理) を高速化するためのクラス