1. **Copying** (sometimes called scavenge): this collector very efficiently moves objects between two or more generations. The source generations are left empty, allowing remaining dead objects to be reclaimed quickly. However, since it requires empty space to operate, copying requires more footprints. **Copying collection** is used for all minor collections.
2. **Mark-compact**: this collector allows generations to be collected in place without reserving extra memory; however, compaction is significantly slower than copying. In 1.3.1 mark-compact is used for major collections.
3. **Incremental** (sometimes called train): this collector is used only if –Xinc. gc is passed on the command line. By careful bookkeeping, incremental GC collects just a portion of the old generation at a time, trying to spread the large pause of a major collection over many minor collections. However, it is even slower than mark-compact when considering overall throughput.