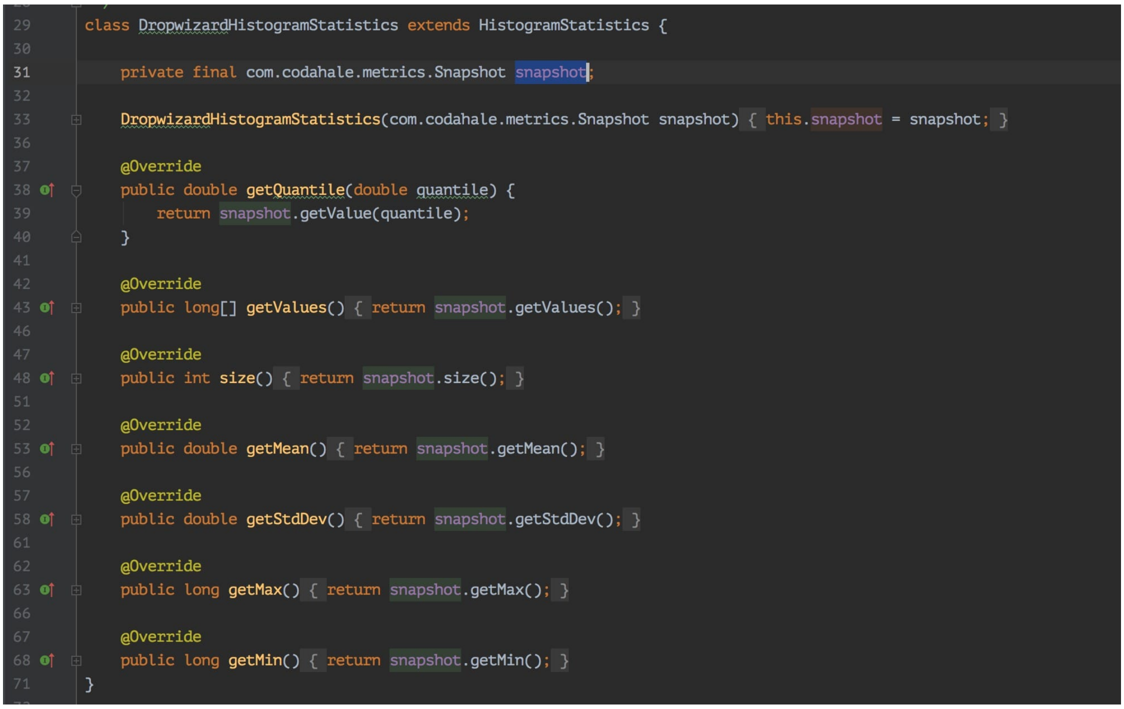
DropwizardHistogramStatistics

实现自 HistogramStatistics 抽象类。该类中有一个属性 snapshot，在类中都通过该属性进行获取对应的值。



```java
class DropwizardHistogramStatistics extends HistogramStatistics {

    private final com.codahale.metrics.Snapshot snapshot;

    DropwizardHistogramStatistics(com.codahale.metrics.Snapshot snapshot) { this.snapshot = snapshot; }

    @Override
    public double getQuantile(double quantile) {
        return snapshot.getValue(quantile);
    }

    @Override
    public long[] getValues() { return snapshot.getValues(); }

    @Override
    public int size() { return snapshot.size(); }

    @Override
    public double getMean() { return snapshot.getMean(); }

    @Override
    public double getStdDev() { return snapshot.getStdDev(); }

    @Override
    public long getMax() { return snapshot.getMax(); }

    @Override
    public long getMin() { return snapshot.getMin(); }
}
```

DropwizardHistogramWrapper

DropwizardHistogramWrapper 实现自 Histogram 接口

DropwizardMeterWrapper

FlinkCounterWrapper

FlinkGaugeWrapper

FlinkHistogramWrapper

FlinkMeterWrapper

HistogramStatisticsWrapper

上面那几个都是包装类

ScheduledDropwizardReporter

该类是一个抽象类，实现了 MetricReporter、Scheduled、Reporter、CharacterFilter 接口，flink-metrics-graphite 模块中的 GraphiteReporter 类就是继承该抽象类。

有这些常量：

```
public static final String ARG_HOST = "host";
public static final String ARG_PORT = "port";
public static final String ARG_PREFIX = "prefix";
public static final String ARG_CONVERSION_RATE = "rateConversion";
public static final String ARG_CONVERSION_DURATION =
"durationConversion";
```

后面需要根据这些常量来进行相关的配置。

除了上面几个常量的话，还有下面这些属性：

```
protected final MetricRegistry registry;

protected ScheduledReporter reporter;

private final Map<Gauge<?>, String> gauges = new HashMap<>();
private final Map<Counter, String> counters = new HashMap<>();
private final Map<Histogram, String> histograms = new HashMap<>();
private final Map<Meter, String> meters = new HashMap<>();
```

重写了 notifyOfAddedMetric 和 notifyOfRemovedMetric 方法，重写 report 方法：

```
public void report() {
    // we do not need to lock here, because the dropwizard registry
is
    // internally a concurrent map
    @SuppressWarnings("rawtypes")
    final SortedMap<String, com.codahale.metrics.Gauge> gauges =
registry.getGauges();
    final SortedMap<String, com.codahale.metrics.Counter> counters
= registry.getCounters();
    final SortedMap<String, com.codahale.metrics.Histogram>
histograms = registry.getHistograms();
    final SortedMap<String, com.codahale.metrics.Meter> meters =
registry.getMeters();
    final SortedMap<String, com.codahale.metrics.Timer> timers =
registry.getTimers();

    this.reporter.report(gauges, counters, histograms, meters,
timers);
}
```