



## JProfiler 内存分析工具说明

### 选项卡说明

**Allocations** 分配情况（快照模式不可用）

**Biggest Object** 大对象

**References** 引用

**Time** 定时器（快照模式不可用）

**Inspections** 检查

**Graph** 图形展示

### Class 选项卡

**shallow size** 浅层大小

**retained size** 保留大小

**deep size** 深层大小

浅层大小不包括引用数组和实例的大小，而只包括相应指针的大小。对于第一个在堆上设置了所有对象的对象，浅的大小等于保留的大小和深层大小。对象本身所占内存大小。

保留的大小计算为浅层大小加上如果要回收当前对象集将被垃圾收集的所有对象的总大小。这个大小告诉你一个对象集后面到底有多少内存。（对象本身 + 仅被该对象直接引用或间接引用对象）的 **Shallow Size**。

深度大小计算为浅大小加上所有引用对象的总大小。在极端情况下，该值可能占整个堆的很大百分比。

**Calculate estimated retained size** 计算估计保留大小

除了元类型数组的值之外，所有保留的大小都是下限。该算法比精确计算快得多，在大多数情况下都足够好。

如果您需要每个类的精确保留大小，请使用感兴趣的类创建一个新的对象集，然后单击顶部“对象集说明”中的“计算保留和深度大小”链接。

### References 选项卡

**incoming reference** 被别的对象引用

**outgoing reference** 引用了别的对象

### Inspections 选项卡

**Duplicate strings**

在当前对象集中查找重复的 `java.lang.string` 对象。

计算完检查之后，您将在所有堆查询器视图的顶部看到一个统计表，在该表中，您可以选择每个重复的字符串值，并分别分析相应的字符串对象。

注意：如果当前对象集中不包含 `java.lang.string` 对象，则检查将返回空对象集。

**Duplicate primitive wrappers**

在当前对象集中查找 `java.lang.integer` 等重复的基本包装对象，计算完检验后，您将在所有堆查看器视图的顶部看到一个统计表，您可以在其中选择每个重复的基本值，并分别分析相应的包装对象。预计起飞时间

注意：如果当前对象集中不包含所选类型的包装对象，则检查将返回空对象集

### Duplicate array

在当前对象集中查找重复的数组。计算完检查之后，您将在所有堆查看器视图的顶部看到一个统计表，在该表中，您可以选择每个重复的数组值，并分别分析相应的数组。

注意：如果当前对象集中不包含所选类型的数组，则检查将返回空的对象集。

### Sparse arrays

查找包含大量 `null` 对象的数组

### Array with zero lenght

查找长度为零的对象数组。这可能是一个使用空值以减少内存消耗。

注意：如果当前对象集中不包含所选类型，则检查将返回空的对象集。

### Hash Maps with bad key distribution

查找键分布不正确的哈希图。这可能是一个对象加速在地图中查找的机会。

在计算出检验结果后，对象组表将根据散列图的分布质量对 `hash map` 进行分组。下面有一个截止点，对象集中不包含哈希映射。具有最高 `hash map` 分发质量的对象将位于顶部

### null fileds

查找具有空值百分比高的字段的对象。这些字段可以移动到派生类或聚合对象。

### object retained by inner class

查找仅由其非静态内部类的上隐式引用的对象。有时这可能是内存不足的原因。

### Objects with many incomming references

查找具有许多 `incomming` 引用

在计算检查之后，对象组表将根据实例的传入引用计数对其进行分组。下面有一个截止点，对象集中不包含任何实例。具有最多传入引用的对象将位于顶部

### Object that reference themselves

查找直接自引用的对象，可能是一个移除字段的机会。

### select weakly referenced objects

查找被 `Weak`、`Soft`、`Phantom` 引用的对象

分析引用路径，检查分析完毕后会跳转到 `References` 选型卡，显示被选择对象的 `GCRoot`

这只适用于在获取堆快照时未删除的弱引用类型。

### remove object retained by weak reference

删除通过弱引用、软引用或虚引用保留的对象。

### select objects referenced by the stack

查找通过 `Java` 栈引用的对象，检查分析完毕后会跳转到 `References` 选型卡，显示被选择对象的 `GCRoot`

### remove objects retained by stack references

删除通过 `Java` 栈引用的对象

### Thread locals

查找被 `ThreadLocal` 所引用的对象

### Classes with identical name

查找名字相同被不同的 **ClassLoader** 加载的类

### Instants group by class loaders

按照 **ClassLoader** 分组查看

按照 **ClassLoader** 查看每一个类加载器所加载的实例数量，分析完毕后可以在 **Class** 选型卡查看

### Instants of classes derived from specific base class

查看从被选择的类或接口所衍生出的类的实例，分析完毕后可以在 **Class** 选项卡查看。

### Http session objects

查看实现了 `javax.servlet.http.HttpSession` 的 **HTTP Session** 对象数量

customer filter 定制过滤器

customer grouping 定制分组显示