Метод equals служит для определения эквивалентности объектов, hashCode служит для определения hash значения объекта, которое должно стремится быть уникальным для неэквивалентных объектов, что бы было как можно меньше коллизий.

Эти методы активно используются стандартными контейнерами HashSet, LinkedHashSet, HashMap, LinkedHashMap.

Использование hashCode обеспечивает высочайшую скорость поиска элементов в контейнерах. Но не исключены коллизии, и когда они наступают, для дальнейшего сравнения элементов используется equals.

|  |
| --- |
| equals должен отвечать следующим условиям  @Override public boolean equals(Object o){...};   1. Для любого x вызов x.equals(x) должен возвращать true; 2. Для любых x и y вызов x.equals(y) должен возвращать true если и только если y.equals(x) возвращает true, то есть всегда x.equals(y) == y.equals(x); 3. Для любых x, y, z если вызов x.equals(y) возвращает true и вызов y.equals(z) возвращает true вызов x.equals(z) должен возвращать true; 4. Для любого x отличного от null вызов x.equals(null) должен возвращать false;   Реализация equals в Object просто сравнивает ссылки, если они совпадают, тогда объекты равны, так что для сравнения разных объектов надо делать свою реализацию. |

|  |
| --- |
| hashCode должен отвечать следующим условиям  @Override public int hashCode(){...};   1. Для любого x вызов x.hashCode() каждый раз должен возвращать одно и тоже hash значение. Если hash значение зависит от изменяемых полей объекта, то это может привести к невозможности поиска объекта в контейнерах использующих hash значение. 2. Нужно стремиться к тому, что бы hash значение для неэквивалентных объектов одного типа было уникальным и равномерно распределенным по диапазону положительного int. 3. При вычислении hash нужно задействовать все поля которые участвуют в вычислении equals что бы эквивалентные объекты имели одинаковое hash значение.   Реализация hashCode в Object находится в native коде.  Рецепт создания своего hashCode находится в книге философия Java в главе 17, абзац переопределение hashCode(). |