



Курс Java-разработчик

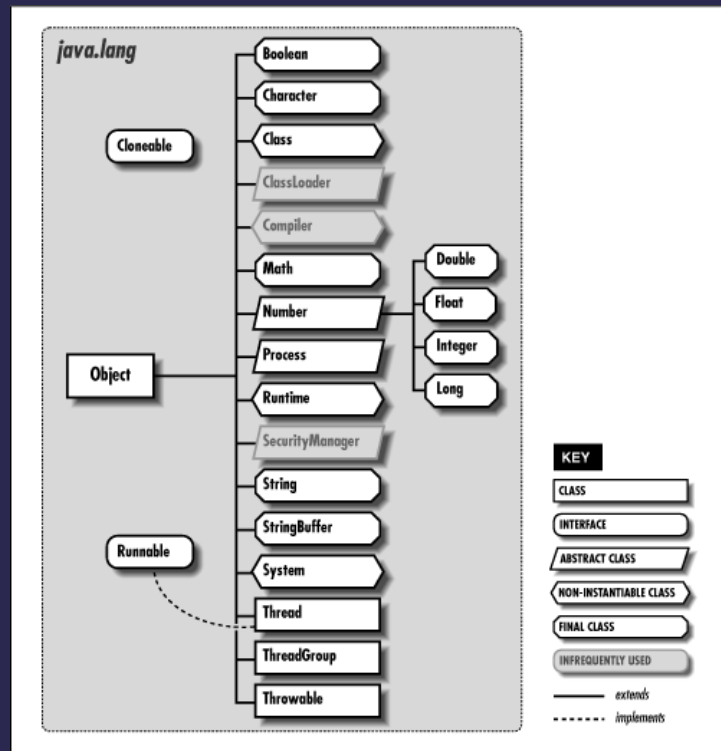


ООП



Класс **Object**

- В Java почти (кроме примитивов) всё является объектами. Класс **Object** является базовым для всех остальных классов.



Методы класса **Object**

- **boolean** equals(**Object** obj);
- **native int** hashCode();
- **String** toString();
- **Object** clone();
- **void** finalize();



equals и hashCode

Метод **hashCode** выдает хэш-код объекта. Хэш-кодом называется уникальный числовой идентификатор, сопоставляемый каждому экземпляру любого класса.

Метод **equals** определяет, являются ли объекты одинаковыми. Если операция `==` проверяет равенство по ссылке (указывают на один и тот же объект), то метод **equals** – равенство по значению (состояния объектов одинаковы).



Contract **equals/hashCode**

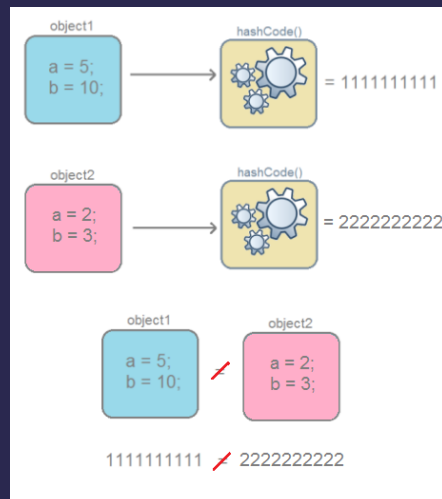
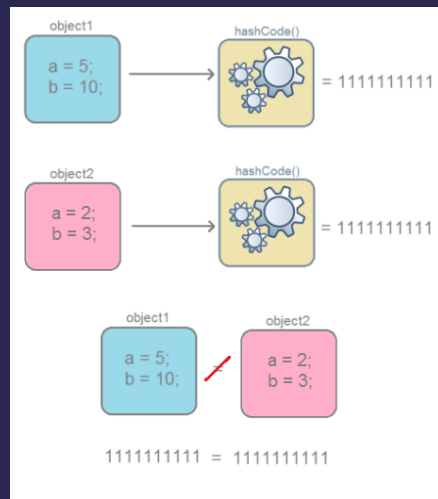
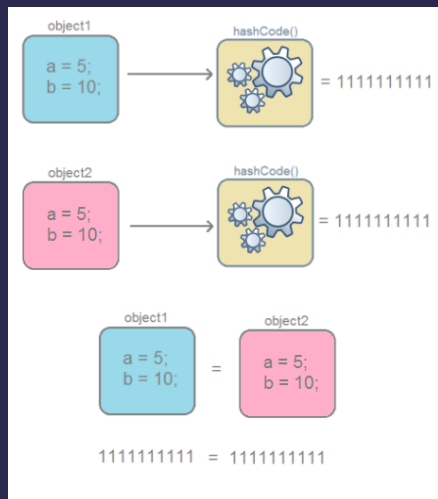
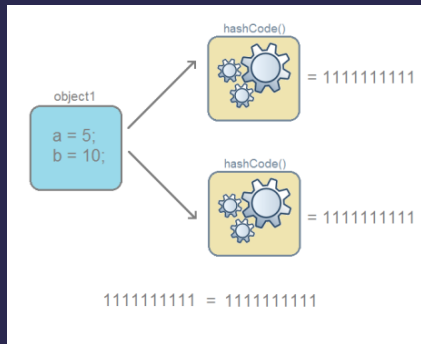
- Если хэш-коды разные, то и входные объекты гарантированно разные.
- Если хэш-коды равны, то входные объекты не всегда равны.

Можно сформулировать в обратном направлении.

- Если объекты равны, то хэш-коды гарантированно равны.
- Если объекты не равны, то хэш-коды не всегда равны.



Contract **equals/hashCode**



Метод **toString**

Возвращает строковое представление объекта.

По умолчанию, вызывает возвращает строку, содержащую название класса объекта и его хэш-код в шестнадцатеричном формате.



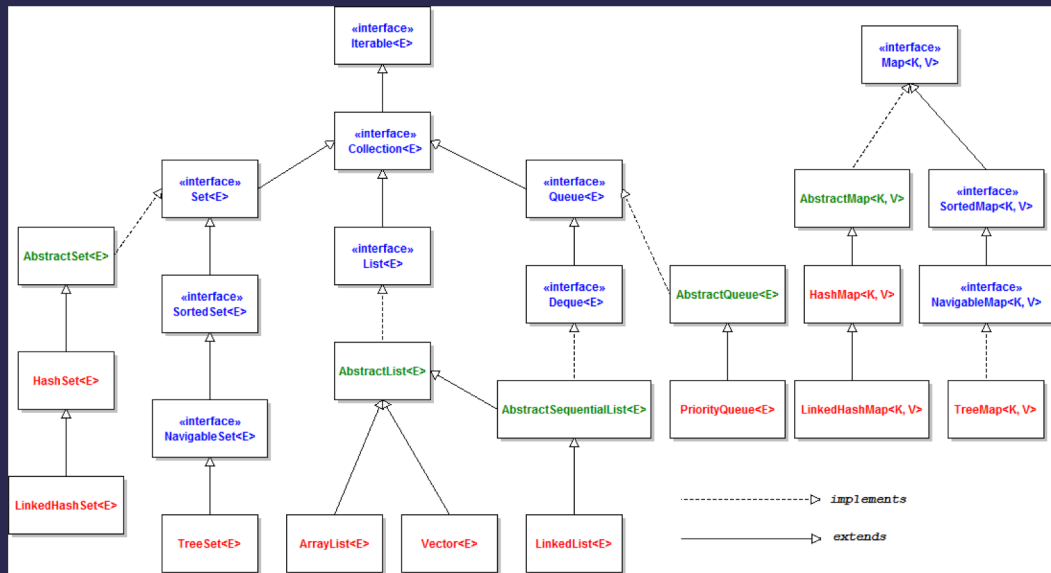


Коллекции



Collection

Контейнерный класс – это класс, объекты которого выступают в роли хранилищ других объектов. Коллекция является неупорядоченной.



ОСНОВНЫЕ МЕТОДЫ **Collection<E>**

- **boolean** add (**E** item)
- **boolean** addAll (**Collection<? extends E>** col)
- **void** clear ()
- **boolean** contains (**Object** item)
- **boolean** isEmpty ()
- **boolean** remove (**Object** item)
- **boolean** removeAll (**Collection<?>** col)
- **int** size ()
- **Object[]** toArray ()



List

Представляет собой неупорядоченную коллекцию, в которой допустимы дублирующие значения. Иногда их называют последовательностями (**sequence**). Элементы такой коллекции пронумерованы, начиная от нуля, к ним можно обратиться по индексу.



ОСНОВНЫЕ МЕТОДЫ **List<E>**

- **void** add(**int** index, **E** obj)
- **boolean** addAll(**int** index, **Collection<? extends E>** col)
- **E** get(**int** index)
- **int** indexOf(**Object** obj)
- **int** lastIndexOf(**Object** obj)
- * **static <E> List<E>** of(**E**... elem)
- **E** remove(**int** index)
- **E** set(**int** index, **E** obj)

* - введен в JDK 9.



Реализации **List**

- **ArrayList** – список на массиве. Длина автоматически увеличивается при добавлении новых элементов. Если при добавлении элемента, оказывается, что массив полностью заполнен, будет создан новый массив размером $(n * 3) / 2 + 1$, в него будут помещены все элементы из старого массива + новый, добавляемый элемент.
- **LinkedList** – двусвязный список. Это структура данных, состоящая из узлов, каждый из которых содержит как собственно данные, так и две ссылки («связки») на следующий и предыдущий узел списка.

