МЕТОДЫ КЛАССА Object:

1. hashCode

Object object = new Object();

Int our\_thing;

our\_thing = object.hashCode();

System.out.print(our\_thing);

Выведеться десятизначное число типа int (2139874612), это и есть битовая строка фиксированной длины

В примере Object является массивом, его обрабатывает метод hash.Code() и дает на выходе int.

Для одного и того же объекта hsCode всегда одинаковый, так что он нужен чтобы понимать, одинаковые ли объекты или нет (в одной ли ячейке памяти содержаться).

Множество возможных hsCode’ов ограничивается областью значений int.

Хэш код - это число, битовая строка, которая имеет фиксированную длину, получена из массива произвольной длины

(это нельзя верить) Если hsCode разные, то и входные объекты разные, а если hsCode одинаковые, то входные объекты не всегда равны.

Коллизия – когда у 2 объектов одинаковые hsкоды, но объекты не равны (подобно ошибке).

Для одного и того же объекта хеш-код всегда одинаковый

Если объекты одинаковые, то и хеш-коды у них одинаковые.

Если хеш-коды равны, то объекты не обязательно равны

Если хеш-коды разные, то и объекты разные

1. getClass

Person ivan = new Person();

System.out.print(ivan.getClass());

Выведет class com.company.Person, ну или в другом случае вывел бы class Person

1. Equals

Person tom = new Person (“Tom”);

Person bob = new Person (“Bob”);

boolean a = tom.equals(bob);

a == false

могу и tom == bob написать, че им не нравиться то?

Equals сравнивает ссылки а не значения, то есть

Person t1 = new Person (“tom”);

Person t2 = new Person (“tom”);

Boolean a = t1.equals(t2);

a == false

Не важно, что у тебя внутри, важно на какой объект в памяти ты указываешь (если сравниваем экземпляры классов)

Числа, например, сравниваются по числовому значению.