

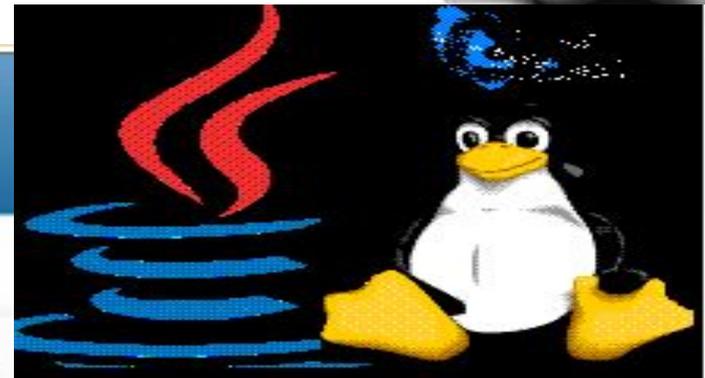
# Java in pictures



## Задачи и результат

### Ключевые идеи

Написанно один раз – работает везде  
Встроенный сборщик мусора  
Безопасность исполнения



### Результат

- + «Переносимые» программисты
- Программы все равно зависят от архитектуры
- + Сборщик мусора устраняет утечки памяти
- Сборщик мусора «всегда» работает в ненужное время

# Java editions



## Java SE (Standard Edition)

Стандартная редакция

JVM + библиотека классов

## Java EE (Enterprise Edition)

Java SE

Спецификация

API + runtime environment services

## Java ME (Micro Edition)

Мобильные телефоны, бытовые приборы

# Типы данных

Примитивы:  
boolean, byte (8)  
char, short, (16)  
int, float, (32)  
long, double (64)

Ссылочные типы

Классы

Интерфейсы

Массивы

enum

bits	type	type
8	byte	boolean
16	short	char
32	int	float
64	long	double

# Программирование.

## Основы синтаксиса языка ЯВУ

### Типы данных. Примитивы.

Байт	8	$-128 \dots +127$	$-2^7 \dots (2^7 - 1)$	byte	
Слово	16	$-32\,768 \dots +32\,767$	$-2^{15} \dots (2^{15} - 1)$	char	short
2 слово	32	$-2\,147\,483\,648 \dots +2\,147\,483\,647$	$-2^{31} \dots (2^{31} - 1)$	int	float
4 слово	64	$-9\,223\,372\,036\,854\,775\,808 \dots +9\,223\,372\,036\,854\,775\,807$	$-2^{63} \dots (2^{63} - 1)$	long	double

*Размеры элементов хранения данных в битах*

Допустимые диапазоны значений целых чисел со знаком

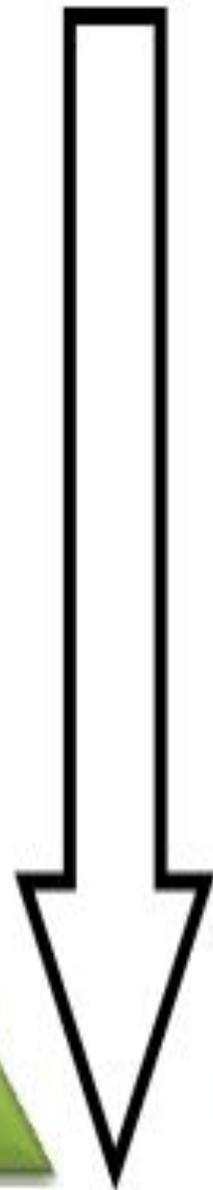
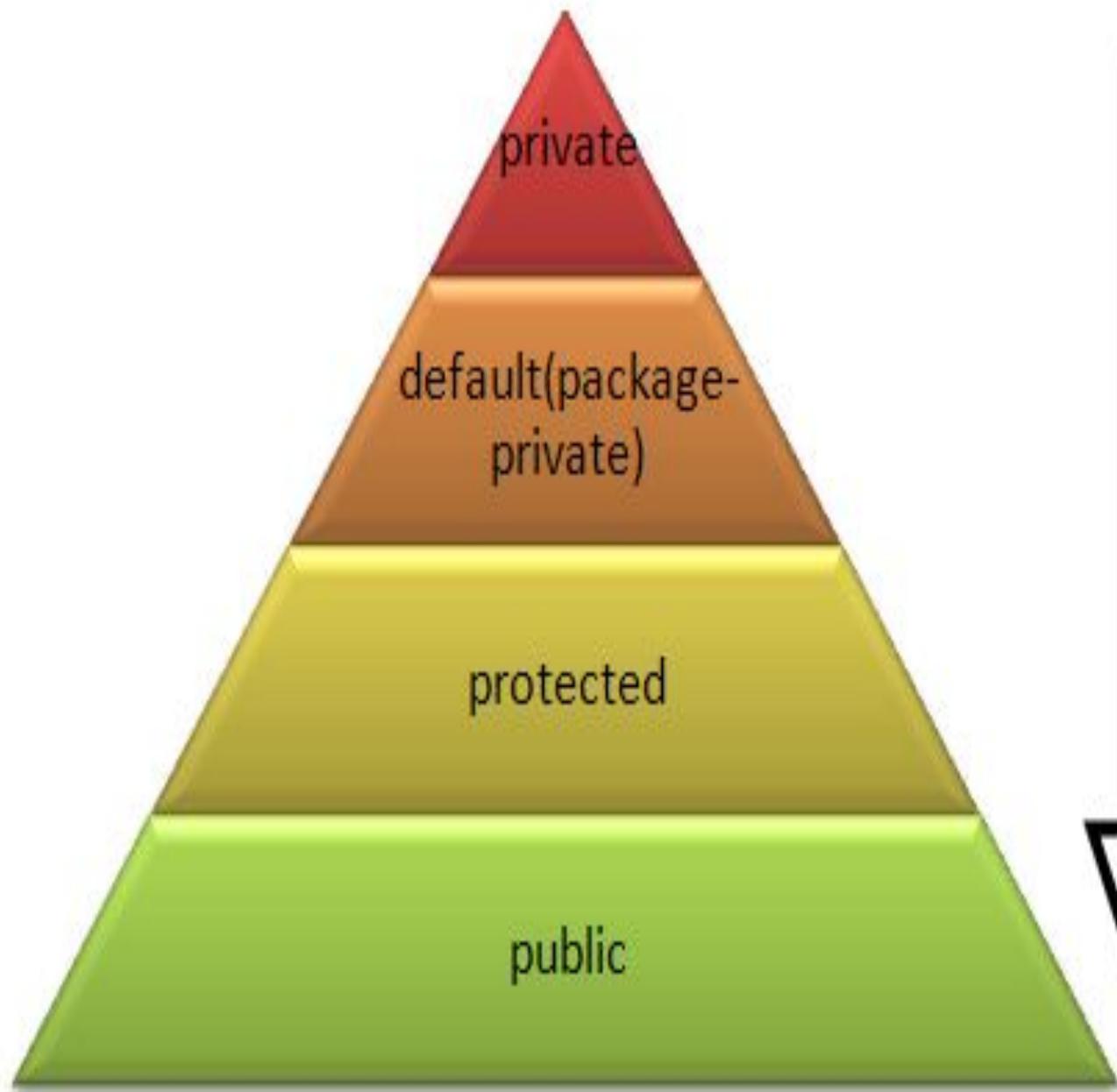
# Autoboxing

## Primitive to Wrapper

- `public class WrapperExample1{`
- `public static void main(String args[]){`
- `//Converting int into Integer`
- `int a=20;`
- `Integer i=Integer.valueOf(a);//converting`
- `Integer j=a;//autoboxing`
- 
- `System.out.println(a+" "+i+" "+j);`
- `}}`

# Wrapper into primitive Unboxing

- public class WrapperExample2{
- public static void main(String args[]){
- //Converting Integer to int
- Integer a=new Integer(3);
- int i=a.intValue();//converting Integer to int
- int j=a;//unboxing, now compiler will write a.intValue() internally
- 
- System.out.println(a+" "+i+" "+j);
- }}



Убывание уровня

закрытости

модификаторов

# Операторы

Условный  
if()...else...

выбора  
switch(){  
case ...:...  
default ...  
}

цикла с  
предусловием  
while(...){}  
}

цикла с  
постусловием  
do{...  
}  
while(...)

параметрического  
цикла  
for(иниц ; услов ;  
шаг){...}

java.lang	базовая функциональность и типы
java.util	коллекции и хелперы
java.io	input / output, работа с файлами
java.math	математические операции
java.security	шифрование, создание ключей
java.net	сеть
java.sql	базы
java.awt	нативная графика
java.swing	платформонезависимая графика
java.applets	апплеты



Duke



# ПРОГРАММИРОВАНИЕ

## PACKAGE JAVA.LANG.;

Class  
Object

Class System

<http://docs.oracle.com/javase/6/docs/api/java/lang/>

String  
StringBuffer  
StringBuilder

Math

Thread

Number  
(Wrappers)

Throwable

Error

Exceptions

# Основные классы в пакете

## java.lang

### Базовый класс Object в java.

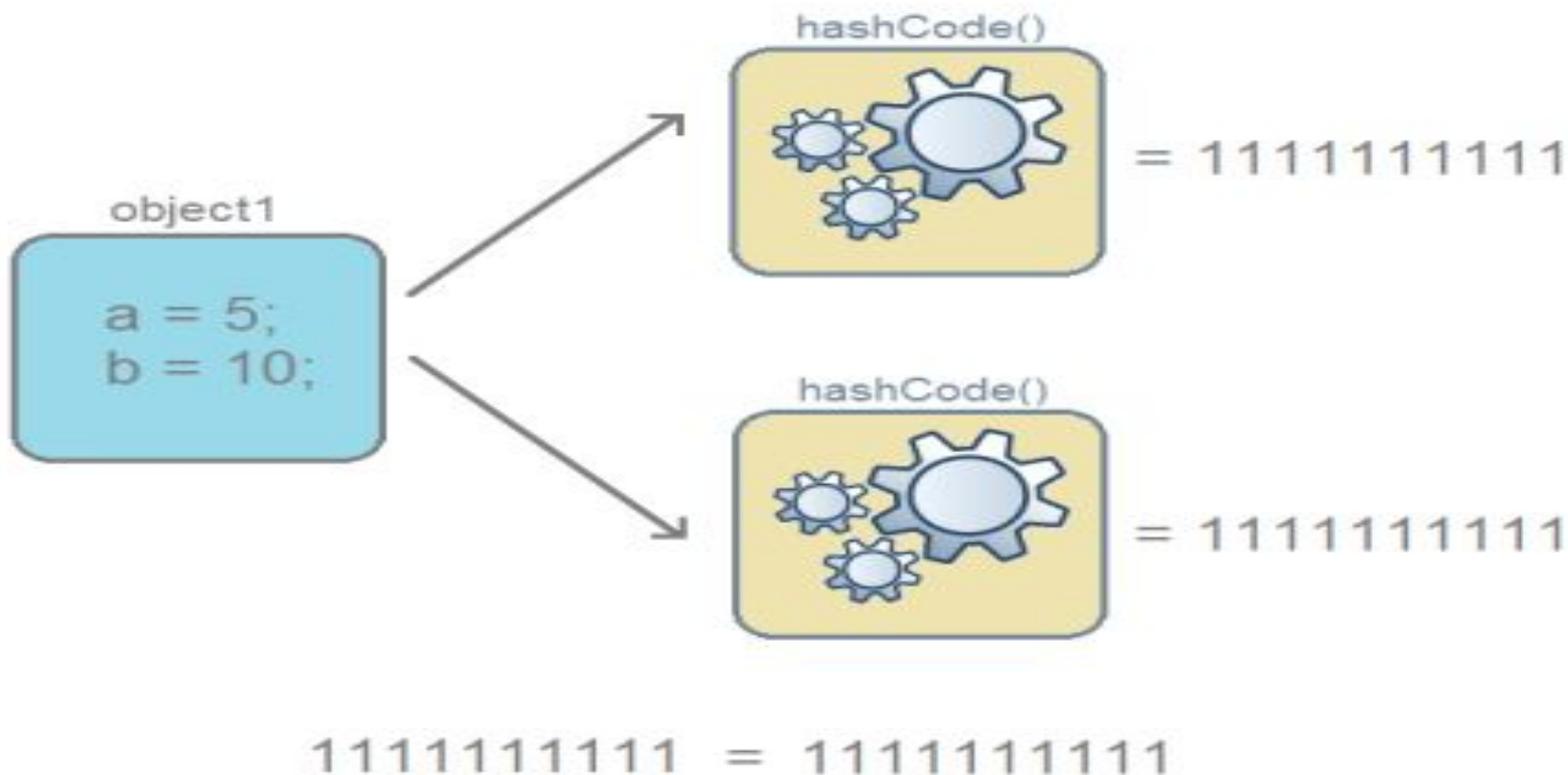
<http://docs.oracle.com/javase/1.4.2/docs/api/java/lang/Object.html>



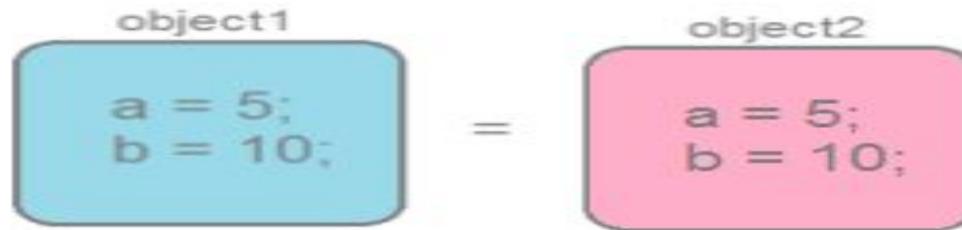
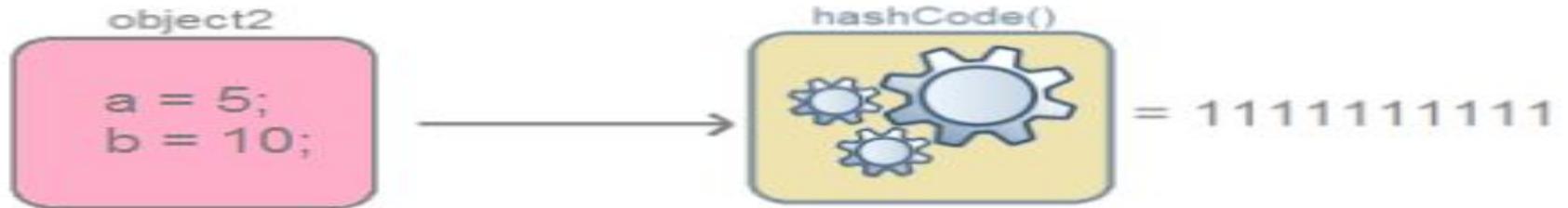
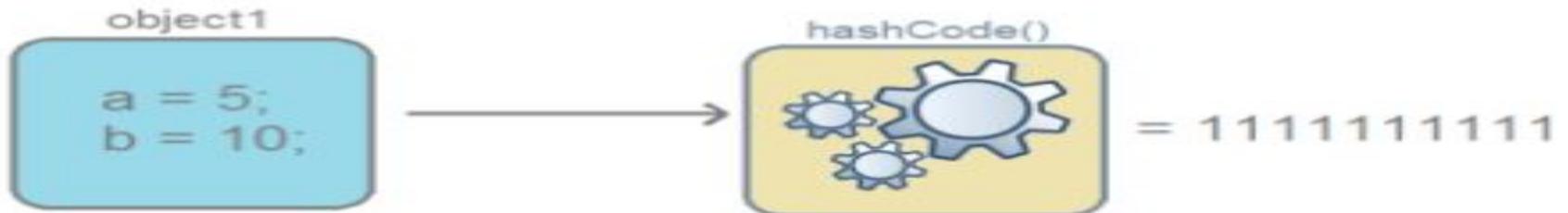
#### Базовый класс Object в java.

- 1. public final native Class getClass()
- 2. public native int hashCode()
- 3. public boolean equals(Object obj)
- 4. protected native Object clone()  
throws  
CloneNotSupportedException
- 5. public String toString()
- 6. public final native void notify()
- 7. public final native void notifyAll()
- 8. public final native void wait(long  
timeout) throws  
InterruptedException
- 9. public final void wait(long timeout,  
int nanos) throws  
InterruptedException
- 10. public final void wait() throws  
InterruptedException
- 11. protected void finalize() throws  
Throwable

для одного и того-же объекта,  
хеш-код всегда будет  
одинаковым;

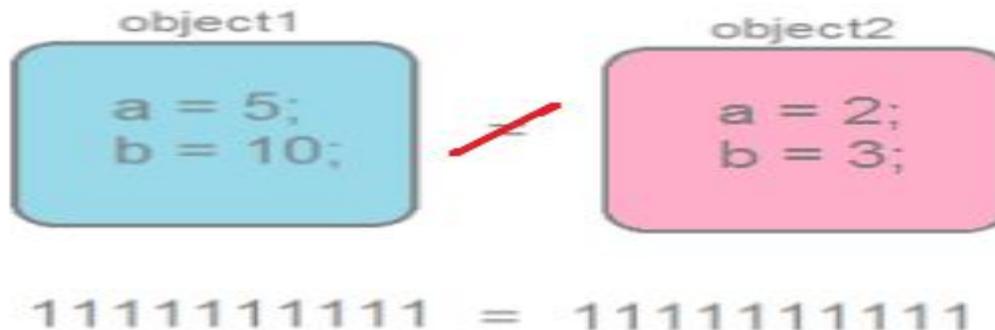
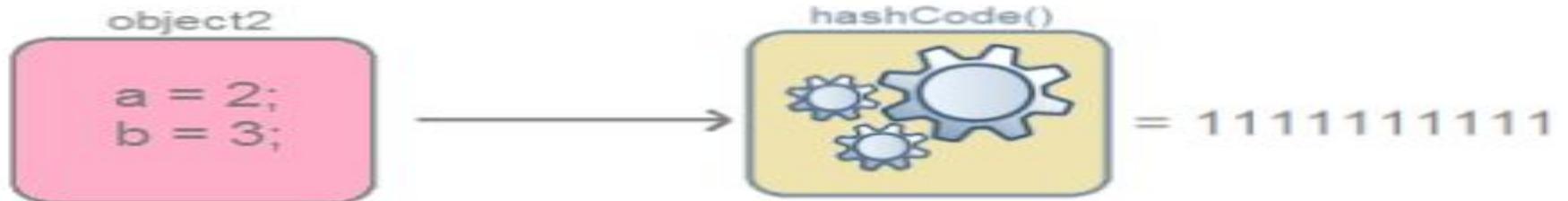
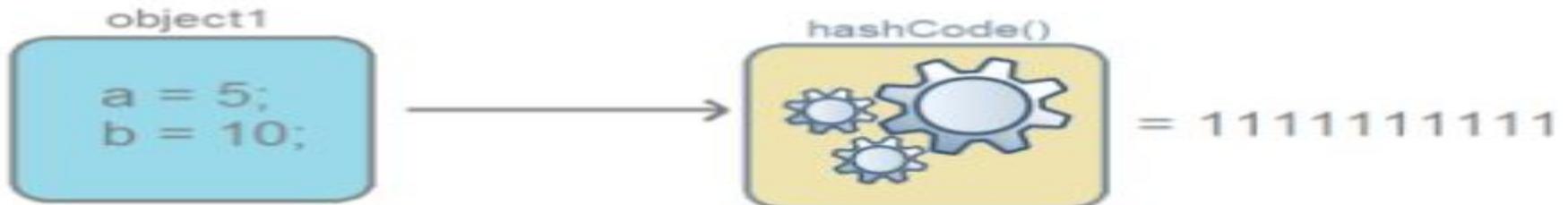


если объекты одинаковые, то и хеш-коды одинаковые (но не наоборот)

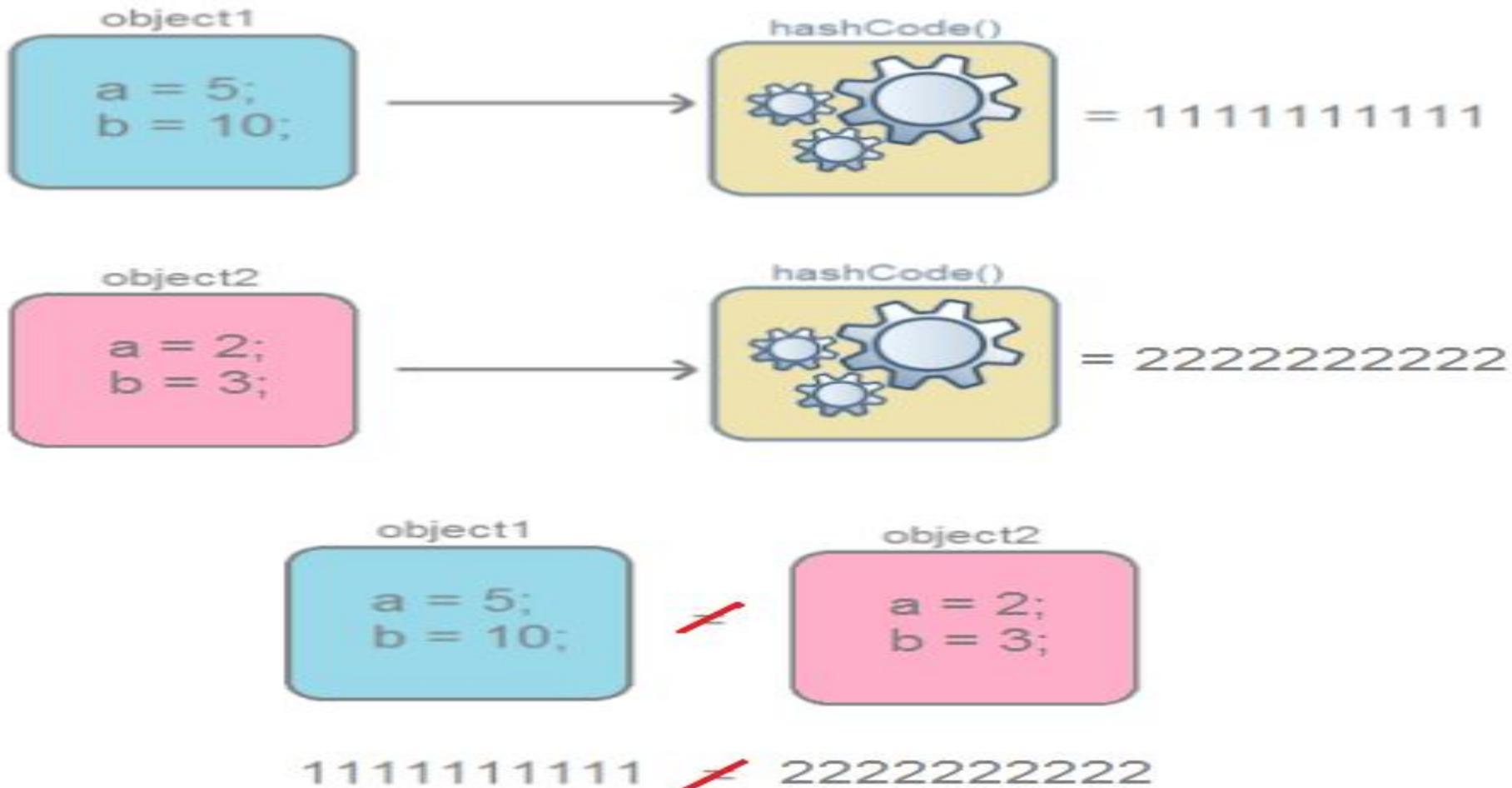


1111111111 = 1111111111

если хеш-коды равны, то  
входные объекты не всегда  
равны (коллизия)



если хеш-коды разные, то и  
объекты гарантированно разные;



# Понятие эквивалентности.

## Метод equals()

- Метод equals() обозначает отношение эквивалентности объектов. Эквивалентным называется отношение, которое является симметричным, транзитивным и рефлексивным.
- **Рефлексивность:** для любого ненулевого x, x.equals(x) вернет true;
- **Транзитивность:** для любого ненулевого x, y и z, если x.equals(y) и y.equals(z) вернет true, тогда и x.equals(z) вернет true;
- **Симметричность:** для любого ненулевого x и y, x.equals(y) должно вернуть true, тогда и только тогда, когда y.equals(x) вернет true.
- Также для любого ненулевого x, x.equals(null) должно вернуть false.  
Отличия equals() от операции == в классе Object нет.

# Class String

39	<code>char[] toCharArray()</code> Converts this string to a new character array.
40	<code>String toLowerCase()</code> Converts all of the characters in this String to lower case using the rules of the default locale.
41	<code>String toLowerCase(Locale locale)</code> Converts all of the characters in this String to lower case using the rules of the given Locale.
42	<code>String toString()</code> This object (which is already a string!) is itself returned.
43	<code>String toUpperCase()</code> Converts all of the characters in this String to upper case using the rules of the default locale.
44	<code>String toUpperCase(Locale locale)</code> Converts all of the characters in this String to upper case using the rules of the given Locale.
45	<code>String trim()</code> Returns a copy of the string, with leading and trailing whitespace omitted.
46	<code>static String valueOf(primitive data type x)</code> Returns the string representation of the passed data type argument.

# Class StringBuffer(неТ у String)

SN	Methods with Description
1	<code>public StringBuffer append(String s)</code> Updates the value of the object that invoked the method. The method takes boolean, char, int, long, Strings etc.
2	<code>public StringBuffer reverse()</code> The method reverses the value of the StringBuffer object that invoked the method.
3	<code>public delete(int start, int end)</code> Deletes the string starting from start index until end index.
4	<code>public insert(int offset, int i)</code> This method inserts an string s at the position mentioned by offset.
5	<code>replace(int start, int end, String str)</code> This method replaces the characters in a substring of this StringBuffer with characters in the specified String.

# Class Number

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

# Основные классы в пакете `java.lang`



## Использование класса `Math` (см. Практический курс)

Класс `Math` является конечным (`final`) и все методы, определённые в классе `Math` являются (статическими) **static**, т. е. невозможно наследовать от класса `Math` и замещать эти методы. Кроме того, класс `Math` имеет приватный конструктор, то есть невозможно создать его экземпляр.

- В классе `Math` есть следующие методы:  
`ceil()`, `floor()`, `max()`, `min()`, `random()`, `abs()`,  
`round()`, `sin()`, `cos()`, `tan()` и `sqrt()`.
  - Метод `ceil()` возвращает наименьшее значение типа `double` (двойной), которое не меньше аргумента и равно математическому целому.
  - Например:
    - `Math.ceil(5.4)` // gives 6
    - `Math.ceil(-6.3)` // gives -6



Duke



# ПРОГРАММИРОВАНИЕ

## PACKAGE JAVA.UUTIL.\*;

Interface  
Collection<E>

Interface  
Comparator  
<T>

Interface  
Iterator<E>

Interface  
Map<K,V>

<http://docs.oracle.com/javase/6/docs/api/java/util/>

Class  
Calendar

Class  
Collections

Class  
Scanner

Class  
StringTokenizer

Class  
Date

Class  
GregorianCalendar

Class  
ResourceBundle

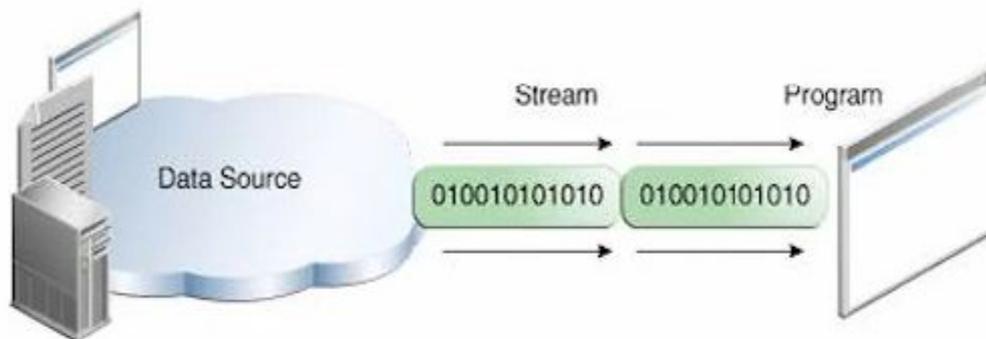
# pacKade java.io:

I/O — общение с внешними устройствами (файлы, принтеры, сеть и т.д.)

Поток — объект который представляет источник или приемник данных

Поток основан на последовательности битов данных

InputStream



OutputStream

