Cheatography

Java Core Classes Cheat Sheet

by Jianmin Feng (taotao) via cheatography.com/79308/cs/19300/

Object class

Universal superclass, all methods should be overriden if default implement is not suitable.

toString()	return memory address by default
equals();	same as == by default, compare memory address, not the content
compareTo()	same as <> by default, memory address instead of content
hashCode()	return unique integer mapping an object to a hash table.
Objects.hash(Object... args)	a simple hash algorithm
getClass()	polymorphic, return class name

ovrridden equals() usually means override hashCode() too to return the same integer if 2 object are the same.

Wrapper class

Wraps (boxing) or unwraps (unboxing) primary types to object type

1. for methods required object type as parameters

2. for java container classes require object items

intObj = Integer (6)	boxing int into Integer
int intObj.intValue()	unboxing

Wrapper object are immutable, final

8 wrapper class

Boolean	Byte	Character	short
Integer	Long	Float	Double

Wrapper class constructor

public Xxxx(xxx)	create from primitive
public Xxxx(String)	create from string

No default constructor;
All wrapper classes are final

Wrapper class methods

Overriden toString(), equals() and compareTo() methods based on object content value instead of memory address.

obj.xxxValue(): used to get primary values

new Xxx(PARAM): Constructor

X.parseXxx(String): parse values from string as primitive

X.valueOf: value from string as Wrapper class

wrapper class equals() check the type first, if not same type return false.

Math

import static java.lang.Math.*;	all members are static
Math.PI	
Math.abs()	
Math.power(base, exp)	
Math.sqrt()	
Math.random	[0,1)
Math.round()	-0.5->0,-0.6->-1, 0.5->1

Math random numbers

a*Math.random()+b	a for scale, b=shift
(int)(Math,random*a)+b	[b,a-1)
(int)((Math,random+1)-*a)+b	[b,a]
(int)(Math,random*-20)+5	[5,24]
(int)(Math,random*-21)+5	[5,25]

Java random number generators

java.lang.Math.random()	double [0,1), need scale,shift,casting
java.Util.Random;	flexible types & methods,reproducible
java.secure.SecureRandom;	<=128bit, cryptographically secure
java.util.concurrent.ThreadLocalRandom;	more efficient with multi-thread app

By **Jianmin Feng** (taotao)
cheatography.com/taotao/

Not published yet.
Last updated 10th May, 2019.
Page 1 of 1.