**Lang pacakage**

Object class in java

Every java class is directly or indirectly derived from the object class.

Import lang.Object; import lang.\*;

System.out.println(obj);

Using Object class method

1) toString()- it provides string representation of object

Class Demo

{

Public static void main(String args[])

{

Demo obj=new Demo();

}

System.out.println(obj);

System.out.println(obj.toString());

}

2. hashcode()-for

3.equals(object obj);

Ex: obj.equals(obj1);

obj.equals(this);

getClass(): obj.getClass(); used to get actual runtime class of the object. It also be used to get metadata of this class.

Finalize();

This method is called just before an object is garbagecollected. It is called by the Garbage collector. We should override finalize() method to dispose system resources, perform clean-up activites and minimize memory leaks.

Clone(): it returns new object that is exactly the same as the object.

->Enumeration:

Ex: enum color

{

Blue,Red,Yellow,Green,Orange;

};

Class Demo21

{

Public static void main(String args[])

{

color ob=color.Red;

color ob=color.Pink;

System.out.println(ob);

}

}

Class Math

->java.lang.Math

Math.abs();

min()

tan()

round()

floor()

cos()

max()

ceil()

avg()

pow()

Java util package->classes & Intefaces

Divided into 2 categories

1) Collection Frame work (collections)

**2) legacy collections**

Collection Frame work collections it got divided into 3 sub categories.

1. 1)core collection interfaces

1.2)general purpose implementations

1.3)more utility collections

1.1) core collection interfaces:

1)collection Interfaces:it is the root interface in the hierarchy

2)List Inteface.

3)Set interface

4)map interface

5)sortedSet interface

6)sortedMap interface

1.2) general purpose implementations

1) ArrayList 2)LinkedList 3)HashSet 4)LinkedHashSet 5)TreeSet 6)HashMap 7)LinkedHashMap 8)TreeMap 9)Priority Queue 10)ArrayQueue.

1.3) more utility collections

1)iterator(interface)

2)ListIterators(interface)

3)Arrays(class)

4)Collections (class)

5)Scanner(class)

2) legacy collections

1. Enumerations(interface)

2. StringTokenizer(class)

3. vector(class)

4) Random

5) Stack

8) Date

9)HashTable

10) Dictionary(abstract class)

Formatter class

The formatter is a built-in class in java used for layout justifications &alignment, Common formats for numeric,string,date/time .The Foramtter class is defined as final class inside the java.util package.

Constructors: 1) Formatter()

2)Formatter(appedable a)

3)Formatter(File file)

Random class: it generates random integers,floating point numbers and Boolean values.

1) nextInteger();

2)nextBytes();

3)nextInteger(int);

import java.util.Random;

//Import java.util.\*;

class Demo25

{

public static void main(String args[])

{

Random obj=new Random();

System.out.println(obj.nextInt());

int x=100;

System.out.println(obj.nextInt(x));

System.out.println(obj.nextBoolean());

}

}

Time package

This package is used for all the date,time related operations , the various classes in this package are

1)Clock

2)Duration

3)Instant

4)LocalDate

5)LocalDateTime

6)MonthDay

7)Period

8)LocalTime

Syntax: import java.time.Instant;

import java.time.\*;

public class {

    public static void main(String[] args)

    {

        // create an Instant object

        Instant lt = Instant.now();

        // print result

        System.out.println("Instant : "

                           + lt);

    }

}

Temporial Adjusters class

import Java time.temporal.TemporaiAdjusters

this class provides Adjusters, which are a key tool for modifying temporal objects.i.e, date like “Second Saturday” of the month or”next Tuesday” etc.

1)invoking the method on the interface directly

2) by using Temporal.with(TemporalAdjuster)

Methods : 1) dayOfWeekInMonth(int ordinal,DayOfWeek dayofweek)

2)firstDayOfMonth()

3)firstDayOfNextMonth()

4)firstDayOfNextYear()

import java.time.DayOfWeek;

import java.time.LocalDate;

import java.time.temporal.TemporalAdjusters;

class Demo29

{

public static void main(String args[])

{

Demo29 obj=new Demo29();

obj.testAdjusters();

}

public void testAdjusters()

{

LocalDate date1=LocalDate.now();

System.out.println("today's date="+date1);

LocalDate nextTuesday=date1.with(TemporalAdjusters.next(DayOfWeek.TUESDAY));

System.out.println("nexttuesday"+nextTuesday);

}

}