**Collection Framework**

Collection Interface--- Divided into two List(Ordered) and Set(Unordered) Interfaces

ArrayList, LinkedList(faster addition and deletion) and Vector

Collection Class

**MAP INTERFACE**

Contains Collections of Object as a unique Key-Value pair. Map is Unordered and doesn’t allow duplicate elements.

1. Map

* Abstract Map-class(I)

-HashMap(E)

--Linked HashMap(E)

* Sorted Map(E)

-Tree Map(I)

**HashMap** **Class**

* Extends Abstract Map implements Map,Clonable,Serializable Interface.
* Contains Collection of Key-Value pair and prints in Random Order.
* Default Capacity is 16.

**Linked Hash Map Class**

* Extends Hashmap, Implements Map,Clonable,Serializable Interface
* Contains Collection of Key-Value pair and prints in the order in which we have inserted.

**Tree Map Class**

* Extends AbstractMap Implements SortedMap,Clonable,Serializable.
* Contains Collection of Key-Value pair and prints in sorted order based on the key.

Set entrySet()🡪Converts both key and values to the Set Interface.

Set keySet()🡪 Converts only keys to the Set Interface.

Map.Entry interface is used to get the key and value separately.

Object getKey()🡪

Object getValue()🡪

**HashTable**

It is a legacy Class, Similar to HashMap but it is synchronized.

enumerationElements()🡪 Converts only values

enumerationKeys()🡪 Converts only Keys

**Properties Class**

* Subclass of HashTable
* It is synchronized
* Collection of Key-Value pairs.
* Print in Random Order but both key and value should be in the String.

**Date Class**

Used to print CurrentDate and Time in Java.

Constructor:

1. Date()🡪 Prints currents date and time
2. Date(long millisecond)🡪 Calculate date and time from Jan 1 1970.

Methods:

1. Boolean after(Date d)🡪
2. Boolean before(Date d)🡪
3. int compareTo(Date d)🡪
4. long getTime()🡪
5. void setTime(long millisecond)🡪

**Calendar Class**

It is an Abstract Class used to extract Useful Information of Date and Time Components like hour, minutes second, month, year, etc.

Constructor:

1. Calendar()

Method:

1. static Calendar getInstance()🡪 Used to Create Object For Calendar Class
2. public void add(int field, int value)🡪
3. public void set(int field, int value)🡪
4. public boolean after(Object o)🡪
5. public boolean before(Object o)🡪
6. public int get(int field)🡪
7. public int getActualMaximum(int field)🡪
8. public int getActualMinimum(int field)🡪
9. public ingetFirstDayOfWeek()🡪
10. public final Date getTime()🡪Converts Calendar to Date
11. public void setTime(Date d)🡪Converts Date to Calendar
12. public int getWeeksInWeekYear()🡪
13. public void setWeekDate(int year, int weekOfYear, int dayOfWeek)🡪
14. public abstract int getMinimum(int field)🡪
15. public abstract int getMaximum(int field)🡪

Constants

1. public static final int hour🡪
2. public static final int Month🡪
3. public static final int Minute🡪
4. public static final int Second🡪
5. public static final int ERA🡪
6. public static final int HOUR\_OF\_DAY🡪
7. public static final int WEEK\_OF\_MONTH🡪
8. public static final int WEEK\_OF\_YEAR🡪
9. public static final int MILLISECOND🡪
10. public static final int DATE🡪
11. public static final int Monday🡪Sunday
12. public static final int January🡪December
13. public static final int DAY\_OF\_WEEK🡪
14. public static final int DAY\_OF\_MONTH🡪

**Gregorian Calendar**

Concurrent implementation of Calendar Class

Constructor

1. GregorianCalendar()
2. GregorianCalendar(int year,int month,int day)
3. GregorianCalendar(int year,int month,int day,int hour,int minute)
4. GregorianCalendar(int year,int month,int day,int hour,int minute,int second)

Method

1. Boolean isLeapYear(int year)

**DATE FORMAT CLASS**

1. It is an abstract class used to format and parse the date in different style
2. Format🡪Date to String
3. Parsing🡪 String to Date
4. Present in java.text.\*; package

Constructor:

DateFormat()

Constant:

1. Public static final int short
2. Public static final int medium
3. Public static final int long
4. Public static final int full

Method:

1. static String format(Date d)
2. dateParse(String s)
3. static DateFormat getInstance()🡪 Prints date and time in short form
4. static DateFormat getDateInstance()🡪 Print only date in medium format
5. static DateFormat getDateInstance(int dateStyle)🡪In date style
6. static DateFormat getTimeInstance()🡪prints time in medium format
7. static DateFormat getTimeInstance(int timeStyle)
8. static DateFormat getDateTimeInstance()🡪prints date and time in medium format
9. static DateFormat getTimeInstance(int dateStyle, int timeStyle)

**SIMPLEDATEFORMAT CLASS**

* Concrete implementation of DateFormat Class Where we can define our own styles.
* Present in java.text.\*;

Constructor

1. SimpleDateFormat()
2. SimpleDateFormat(String pattern)🡪

* GG🡪Represents era(AD,BC) in text
* Y🡪 Represents Year in text
* M🡪 Represents month in text
* MMMM🡪January|MMM-Jan|MM-01
* w🡪 Represents week in number
* W🡪 Week in month in number
* D🡪 Represents Day in year in number
* D🡪Day in m onth in number
* H🡪 hours in day
* M🡪minutes
* S-->milliseconds
* Z🡪Timezone