

Guía de Referencia de la API de Java

Leyenda: ■ Clase
▷ Método de Instancia
• Constructor

□ Interfaz
▷ Método de Clase
→ Herencia

■ java.lang.Object

▷ String toString();
▷ boolean equals(Object);
▷ int hashCode();
▷ Object clone();
▷ void finalize();
▷ Class getClass();

■ enum Tipo { Val1,...,ValN }

▷ Tipo valueOf(String);
▷ Tipo[] values();
▷ int compareTo(Tipo);
▷ int ordinal();

■ java.lang.Throwable

▷ String getMessage();
▷ void printStackTrace();

■ java.lang.System

▷ void arraycopy(T[],int,T[],int,int);
▷ long currentTimeMillis();

■ java.lang.Math

▷ T abs(T);
▷ double cos(double);
▷ double exp(double);
▷ double hypot(double,double);
▷ T max(T,T);
▷ T min(T,T);
▷ double pow(double,double);
▷ double random();
▷ int round(double);
▷ double sin(double);
▷ double sqrt(double);
▷ double tan(double);

■ java.lang.Boolean

▷ int compare(boolean,boolean);
▷ int hashCode(boolean);
▷ boolean parseBoolean(String);
▷ String toString(boolean);
▷ Boolean valueOf(String);

■ java.lang.Integer

▷ int compare(int,int);
▷ int hashCode(int);
▷ int parseInt(String);
▷ String toString(int);
▷ Integer valueOf(String);

■ java.lang.Double

▷ int compare(double,double);
▷ int hashCode(double);
▷ double parseDouble(String);
▷ String toString(double);
▷ Double valueOf(String);

■ java.lang.Character

▷ int compare(char,char);
▷ int hashCode(char);
▷ String toString(char);
▷ boolean isDigit(char);
▷ boolean isLetter(char);
▷ boolean isLowerCase(char);
▷ boolean isUpperCase(char);
▷ boolean isSpaceChar(char);
▷ char toLowerCase(char);
▷ char toUpperCase(char);

■ java.util.Random

▷ int nextInt(int);
▷ int nextInt();
▷ double nextDouble();
▷ double nextGaussian();

■ java.lang.String

▷ String format("%b %c %d %f %s\n",...); // Locale.FRENCH
▷ String format(Locale, "%b %c %d %f %s\n",...); // Locale.ENGLISH
▷ String valueOf(T);
▷ char charAt(int);
▷ int compareTo(String);
▷ int compareToIgnoreCase(String);
▷ String concat(String); // + +=
▷ boolean contains(String);
▷ boolean equalsIgnoreCase(String);
▷ int indexOf(char/String);
▷ int lastIndexOf(char/String);
▷ int length();
▷ String replace(String,String);
▷ String replaceFirst(String,String);
▷ String[] split(String);
▷ String[] split(String,int);
▷ String substring(int);
▷ String substring(int,int);
▷ String toLowerCase();
▷ String toUpperCase();
▷ String trim();

■ java.lang.StringBuilder

▷ int length();
▷ char charAt(int);
▷ int indexOf(char/String);
▷ int lastIndexOf(char/String);
▷ String substring(int);
▷ String substring(int,int);
▷ StringBuilder append(String);
▷ StringBuilder delete(int,int);
▷ StringBuilder deleteCharAt(int);
▷ StringBuilder insert(int,String);
▷ StringBuilder replace(int,int,String);
▷ StringBuilder reverse();
▷ StringBuilder setCharAt(int,char);

■ java.util.StringJoiner

• StringJoiner(String,String,String);
▷ StringJoiner add(String);

■ java.util.Scanner

• Scanner(String);
• Scanner(InputStream);
• Scanner(File);
▷ void close();
▷ boolean hasNextLine();
▷ boolean hasNext();
▷ boolean hasNextInt();
▷ boolean hasNextDouble();
▷ String nextLine();
▷ String next();
▷ int nextInt();
▷ double nextDouble();
▷ Scanner skip(String);
▷ Scanner useDelimiter(String);
▷ Scanner useLocale(Locale); // Locale.ENGLISH , Locale.FRENCH

■ java.util.Arrays

▷ List<T> asList(T[]);
▷ int binarySearch(T[],T);
▷ T[] copyOf(T[],int);
▷ T[] copyOfRange(T[],int,int);
▷ boolean equals(T[],T[]);
▷ void fill(T[],T);
▷ int hashCode(T[]);
▷ void sort(T[]);
▷ String toString(T[]);

■ java.util.Optional<T>

▷ Optional<T> empty();
▷ Optional<T> of(T);
▷ boolean isPresent();
▷ T get();
▷ T orElse(T);

```

■ java.io.File
• File(String);
• File(String,String);
• File(File,String);
▶ boolean exists();
▶ boolean delete();
▶ boolean mkdir();
▶ boolean mkdirs();
▶ boolean renameTo(File);

■ java.io.PrintWriter
• PrintWriter(String);
• PrintWriter(File);
• PrintWriter(Writer);
• PrintWriter(OutputStream,boolean);
▶ void close();
▶ void flush();
▶ void print(String);
▶ void println(String);
▶ PrintWriter printf(String,...);
▶ PrintWriter printf(Locale,String,...);

□ java.lang.Comparable<T>
▶ int compareTo(T);

□ java.util.Comparator<T>
▶ Comparator<T> naturalOrder();
▶ Comparator<T> reverseOrder();
▶ int compare(T,T);
▶ Comparator<T> reversed(){default}
▶ Comparator<T> thenComparing(Comparator<T>){default}

■ java.util.Collections
▶ T min(Collection<T>);
▶ T min(Collection<T>,Comparator<T>);
▶ T max(Collection<T>);
▶ T max(Collection<T>,Comparator<T>);
▶ int binarySearch(List<T>,T);
▶ int binarySearch(List<T>,T,Comparator<T>);
▶ void copy(List<T>,List<T>);
▶ void fill(List<T>,T);
▶ void reverse(List<T>);
▶ void shuffle(List<T>);
▶ void sort(List<T>);
▶ void sort(List<T>,Comparator<T>);

□ java.util.Collection<T> → Iterable<T>
▶ int size();
▶ boolean isEmpty();
▶ boolean contains(Object);
▶ boolean containsAll(Collection<T>);
▶ boolean add(T);
▶ boolean addAll(Collection<T>);
▶ void clear();
▶ boolean remove(Object);
▶ boolean removeAll(Collection<T>);
▶ boolean retainAll(Collection<T>);
▶ Iterator<T> iterator();
▶ Object[] toArray();
▶ T[] toArray(T[]);

□ java.util.Iterator<T>
▶ boolean hasNext();
▶ T next();
▶ void remove();

□ java.util.ListIterator<T> → Iterator<T>
▶ boolean hasNext();
▶ boolean hasPrevious();
▶ T next();
▶ T previous();
▶ int nextIndex();
▶ int previousIndex();
▶ void add(T);
▶ void remove();
▶ void set(T);

■ (C) java.lang.Exception → Throwable
■ (C) java.io.IOException → Exception
■ (C) java.io.FileNotFoundException → IOException

□ java.util.Set<T> → Collection<T>
▶ Set<T> of(T...);

□ java.util.SortedSet<T> → Set<T>
▶ T first();
▶ T last();
▶ Comparator<T> comparator();
▶ SortedSet<T> headSet(T);
▶ SortedSet<T> tailSet(T);
▶ SortedSet<T> subSet(T,T);

□ java.util.List<T> → Collection<T>
▶ List<T> of(T...);
▶ T get(int);
▶ int indexOf(Object);
▶ int lastIndexOf(Object);
▶ void add(int,T);
▶ boolean add(T);
▶ boolean addAll(int,Collection<T>);
▶ T remove(int);
▶ T set(int,T);
▶ void sort(Comparator<T>);
▶ ListIterator<T> listIterator();
▶ ListIterator<T> listIterator(int);
▶ List<T> subList(int,int);

□ java.util.Queue<T> → Collection<T>
▶ T element(); // throws
▶ T peek();
▶ void add(T); // throws
▶ boolean offer(T);
▶ T remove(); // throws
▶ T poll();

□ java.util.Map<K,V>
▶ Map<K,V> of(K,V,...);
▶ Map.Entry<K,V> entry(K,V);
▶ Map<K,V> ofEntries(Map.Entry<K,V>...);
▶ int size();
▶ boolean isEmpty();
▶ V get(Object);
▶ V getOrDefault(Object,V);
▶ boolean containsKey(Object);
▶ boolean containsValue(Object);
▶ V put(K,V);
▶ V putIfAbsent(K,V);
▶ void putAll(Map<K,V>);
▶ void clear();
▶ V remove(Object);
▶ Set<K> keySet();
▶ Collection<V> values();
▶ Set<Map.Entry<K,V>> entrySet();

□ java.util.Map.Entry<K,V>
▶ K getKey();
▶ V getValue();
▶ V setValue(V);

□ java.util.SortedMap<K,V> → Map<K,V>
▶ K firstKey();
▶ K lastKey();
▶ Comparator<K> comparator();
▶ SortedMap<K,V> headMap(K);
▶ SortedMap<K,V> tailMap(K);
▶ SortedMap<K,V> subMap(K,K);

□ java.awt.event.ActionListener
▶ void actionPerformed(ActionEvent);

■ java.awt.event.ActionEvent
▶ String getActionCommand();
▶ Object getSource();

■ (N) java.lang.RuntimeException → Exception
■ (N) java.lang.ArithmeticException → RuntimeException
■ (N) java.lang.IndexOutOfBoundsException → RuntimeException
■ (N) java.lang.NullPointerException → RuntimeException
■ (N) java.lang.IllegalArgumentException → RuntimeException
■ (N) java.lang.NumberFormatException → RuntimeException
■ (N) java.util.NoSuchElementException → RuntimeException
■ (N) java.util.InputMismatchException → NoSuchElementException

```