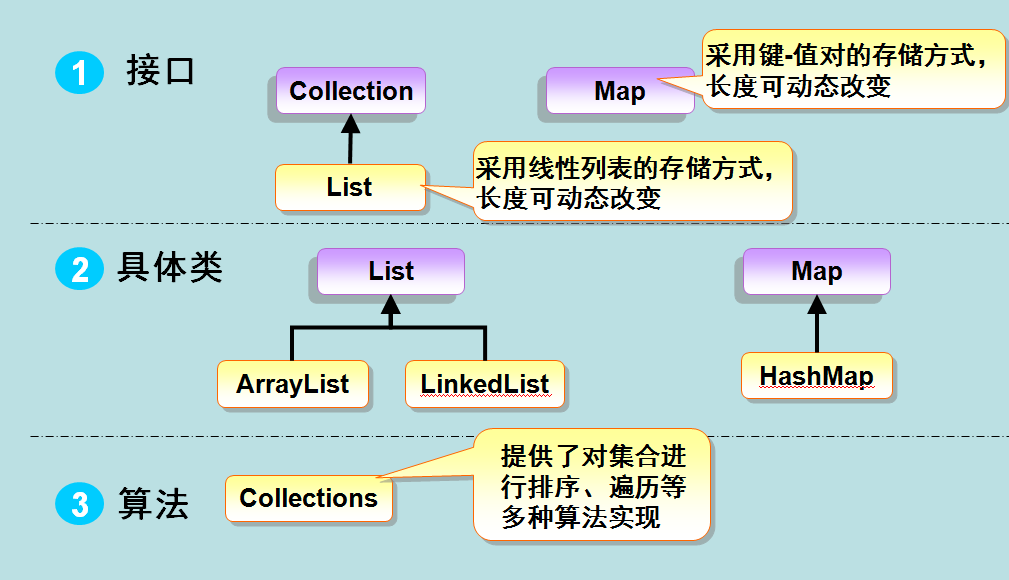
BitSet中实现了Cloneable接口中定义的方法如下表所列：

|  |  |
| --- | --- |
| **序号** | **方法描述** |
| 1 | void and(BitSet set) 对此目标位 set 和参数位 set 执行逻辑与操作。 |
| 2 | void andNot(BitSet set) 清除此 BitSet 中所有的位，其相应的位在指定的 BitSet 中已设置。 |
| 3 | int cardinality( ) 返回此 BitSet 中设置为 true 的位数。 |
| 4 | void clear( ) 将此 BitSet 中的所有位设置为 false。 |
| 5 | void clear(int index) 将索引指定处的位设置为 false。 |
| 6 | void clear(int startIndex, int endIndex) 将指定的 startIndex（包括）到指定的 toIndex（不包括）范围内的位设置为 false。 |
| 7 | Object clone( ) 复制此 BitSet，生成一个与之相等的新 BitSet。 |
| 8 | boolean equals(Object bitSet) 将此对象与指定的对象进行比较。 |
| 9 | void flip(int index) 将指定索引处的位设置为其当前值的补码。 |
| 10 | void flip(int startIndex, int endIndex) 将指定的 fromIndex（包括）到指定的 toIndex（不包括）范围内的每个位设置为其当前值的补码。 |
| 11 | boolean get(int index) 返回指定索引处的位值。 |
| 12 | BitSet get(int startIndex, int endIndex) 返回一个新的 BitSet，它由此 BitSet 中从 fromIndex（包括）到 toIndex（不包括）范围内的位组成。 |
| 13 | int hashCode( ) 返回此位 set 的哈希码值。 |
| 14 | boolean intersects(BitSet bitSet) 如果指定的 BitSet 中有设置为 true 的位，并且在此 BitSet 中也将其设置为 true，则返回 true。 |
| 15 | boolean isEmpty( ) 如果此 BitSet 中没有包含任何设置为 true 的位，则返回 true。 |
| 16 | int length( ) 返回此 BitSet 的"逻辑大小"：BitSet 中最高设置位的索引加 1。 |
| 17 | int nextClearBit(int startIndex) 返回第一个设置为 false 的位的索引，这发生在指定的起始索引或之后的索引上。 |
| 18 | int nextSetBit(int startIndex) 返回第一个设置为 true 的位的索引，这发生在指定的起始索引或之后的索引上。 |
| 19 | void or(BitSet bitSet) 对此位 set 和位 set 参数执行逻辑或操作。 |
| 20 | void set(int index) 将指定索引处的位设置为 true。 |
| 21 | void set(int index, boolean v)  将指定索引处的位设置为指定的值。 |
| 22 | void set(int startIndex, int endIndex) 将指定的 fromIndex（包括）到指定的 toIndex（不包括）范围内的位设置为 true。 |
| 23 | void set(int startIndex, int endIndex, boolean v) 将指定的 fromIndex（包括）到指定的 toIndex（不包括）范围内的位设置为指定的值。 |
| 24 | int size( ) 返回此 BitSet 表示位值时实际使用空间的位数。 |
| 25 | String toString( ) 返回此位 set 的字符串表示形式。 |
| 26 | void xor(BitSet bitSet) 对此位 set 和位 set 参数执行逻辑异或操作。 |

 Vector 还定义了以下方法：

|  |  |
| --- | --- |
| **序号** | **方法描述** |
| 1 | void add(int index, Object element)   在此向量的指定位置插入指定的元素。 |
| 2 | boolean add(Object o)   将指定元素添加到此向量的末尾。 |
| 3 | boolean addAll(Collection c)  将指定 Collection 中的所有元素添加到此向量的末尾，按照指定 collection 的迭代器所返回的顺序添加这些元素。 |
| 4 | boolean addAll(int index, Collection c)  在指定位置将指定 Collection 中的所有元素插入到此向量中。 |
| 5 | void addElement(Object obj)   将指定的组件添加到此向量的末尾，将其大小增加 1。 |
| 6 | int capacity()  返回此向量的当前容量。 |
| 7 | void clear()  从此向量中移除所有元素。 |
| 8 | Object clone()  返回向量的一个副本。 |
| 9 | boolean contains(Object elem)  如果此向量包含指定的元素，则返回 true。 |
| 10 | boolean containsAll(Collection c)  如果此向量包含指定 Collection 中的所有元素，则返回 true。 |
| 11 | void copyInto(Object[] anArray)   将此向量的组件复制到指定的数组中。 |
| 12 | Object elementAt(int index)  返回指定索引处的组件。 |
| 13 | Enumeration elements()  返回此向量的组件的枚举。 |
| 14 | void ensureCapacity(int minCapacity)  增加此向量的容量（如有必要），以确保其至少能够保存最小容量参数指定的组件数。 |
| 15 | boolean equals(Object o)  比较指定对象与此向量的相等性。 |
| 16 | Object firstElement()  返回此向量的第一个组件（位于索引 0) 处的项）。 |
| 17 | Object get(int index)  返回向量中指定位置的元素。 |
| 18 | int hashCode()  返回此向量的哈希码值。 |
| 19 | int indexOf(Object elem)   返回此向量中第一次出现的指定元素的索引，如果此向量不包含该元素，则返回 -1。 |
| 20 | int indexOf(Object elem, int index)   返回此向量中第一次出现的指定元素的索引，从 index 处正向搜索，如果未找到该元素，则返回 -1。 |
| 21 | void insertElementAt(Object obj, int index)  将指定对象作为此向量中的组件插入到指定的 index 处。 |
| 22 | boolean isEmpty()  测试此向量是否不包含组件。 |
| 23 | Object lastElement()  返回此向量的最后一个组件。 |
| 24 | int lastIndexOf(Object elem)   返回此向量中最后一次出现的指定元素的索引；如果此向量不包含该元素，则返回 -1。 |
| 25 | int lastIndexOf(Object elem, int index)  返回此向量中最后一次出现的指定元素的索引，从 index 处逆向搜索，如果未找到该元素，则返回 -1。 |
| 26 | Object remove(int index)   移除此向量中指定位置的元素。 |
| 27 | boolean remove(Object o)  移除此向量中指定元素的第一个匹配项，如果向量不包含该元素，则元素保持不变。 |
| 28 | boolean removeAll(Collection c)  从此向量中移除包含在指定 Collection 中的所有元素。 |
| 29 | void removeAllElements()  从此向量中移除全部组件，并将其大小设置为零。 |
| 30 | boolean removeElement(Object obj)  从此向量中移除变量的第一个（索引最小的）匹配项。 |
| 31 | void removeElementAt(int index)  删除指定索引处的组件。 |
| 32 | protected void removeRange(int fromIndex, int toIndex) 从此 List 中移除其索引位于 fromIndex（包括）与 toIndex（不包括）之间的所有元素。 |
| 33 | boolean retainAll(Collection c)  在此向量中仅保留包含在指定 Collection 中的元素。 |
| 34 | Object set(int index, Object element)  用指定的元素替换此向量中指定位置处的元素。 |
| 35 | void setElementAt(Object obj, int index)  将此向量指定 index 处的组件设置为指定的对象。 |
| 36 | void setSize(int newSize)   设置此向量的大小。 |
| 37 | int size()   返回此向量中的组件数。 |
| 38 | List subList(int fromIndex, int toIndex)  返回此 List 的部分视图，元素范围为从 fromIndex（包括）到 toIndex（不包括）。 |
| 39 | Object[] toArray()  返回一个数组，包含此向量中以恰当顺序存放的所有元素。 |
| 40 | Object[] toArray(Object[] a)  返回一个数组，包含此向量中以恰当顺序存放的所有元素；返回数组的运行时类型为指定数组的类型。 |
| 41 | String toString()  返回此向量的字符串表示形式，其中包含每个元素的 String 表示形式。 |
| 42 | void trimToSize()    对此向量的容量进行微调，使其等于向量的当前大小。 |



下图显示了一个线程完整的生命周期。



**Thread 方法**

下表列出了Thread类的一些重要方法：

|  |  |
| --- | --- |
| **序号** | **方法描述** |
| 1 | **public void start()** 使该线程开始执行；**Java** 虚拟机调用该线程的 run 方法。 |
| 2 | **public void run()** 如果该线程是使用独立的 Runnable 运行对象构造的，则调用该 Runnable 对象的 run 方法；否则，该方法不执行任何操作并返回。 |
| 3 | **public final void setName(String name)** 改变线程名称，使之与参数 name 相同。 |
| 4 | **public final void setPriority(int priority)**  更改线程的优先级。 |
| 5 | **public final void setDaemon(boolean on)** 将该线程标记为守护线程或用户线程。 |
| 6 | **public final void join(long millisec)** 等待该线程终止的时间最长为 millis 毫秒。 |
| 7 | **public void interrupt()** 中断线程。 |
| 8 | **public final boolean isAlive()** 测试线程是否处于活动状态。 |

测试线程是否处于活动状态。 上述方法是被Thread对象调用的。下面的方法是Thread类的静态方法。

|  |  |
| --- | --- |
| **序号** | **方法描述** |
| 1 | **public static void yield()** 暂停当前正在执行的线程对象，并执行其他线程。 |
| 2 | **public static void sleep(long millisec)** 在指定的毫秒数内让当前正在执行的线程休眠（暂停执行），此操作受到系统计时器和调度程序精度和准确性的影响。 |
| 3 | **public static boolean holdsLock(Object x)** 当且仅当当前线程在指定的对象上保持监视器锁时，才返回 true。 |
| 4 | **public static Thread currentThread()** 返回对当前正在执行的线程对象的引用。 |
| 5 | **public static void dumpStack()** 将当前线程的堆栈跟踪打印至标准错误流。 |