|  |  |  |
| --- | --- | --- |
| BagInterface | ArrayBag | ArrayBagDemo |
|  | private final T[] bag;  private int numberOfEntries; private boolean initialized = false; private static final int DEFAULT\_CAPACITY = 25; private static final int MAX\_CAPACITY = 10000; |  |
|  | public ArrayBag()  {  this(DEFAULT\_CAPACITY); }  public ArrayBag(int desiredCapacity){} |  |
|  |  |  |
| public int getCurrentSize(); |  |  |
| public boolean isEmpty(); | public boolean isEmpty(){} | private static void testIsEmpty(BagInterface<String> aBag, boolean correctResult){} |
| public boolean add(T newEntry); | public boolean add(T newEntry){} |  |
| public T remove(); | public T remove(){} | private static void testRemove(BagInterface<String> aBag, String[] tests){} |
| public boolean remove(T anEntry); | public boolean remove(T anEntry){} |  |
| public void clear(); | public void clear(){} |  |
| public int getFrequencyOf(T anEntry); | public int getFrequencyOf(T anEntry){} | private static void testFrequency(BagInterface<String> aBag, String[] tests){} |
| public boolean contains(T anEntry); | public boolean contains(T anEntry){} | private static void testContains(BagInterface<String> aBag, String[] tests){} |
| public T[] toArray(); | public T[] toArray(){} |  |
| // public BagInterface<T> union(BagInterface<T> anotherBag); |  |  |
| public BagInterface<T> intersection(BagInterface<T> anotherBag); |  |  |
| public BagInterface<T> difference(BagInterface<T> anotherBag); |  |  |
|  | private boolean isArrayFull(){} |  |
|  | private int getIndexOf(T anEntry){} |  |
|  | private T removeEntry(int givenIndex){} |  |
|  | private void checkInitialization() {} |  |
|  |  | public static void main(String[] args)  {} |
|  |  | private static void displayBag(BagInterface<String> aBag){} |