```java
package edu.wsu.gridstat.subscriber.interfaces;

/**
 * <p>Title: HolderBaseWithDVInterface</p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2005</p>
 * <p>Company: Washington State University</p>
 * @author Kjell Harald Gjermundrod
 * @version 1.0
 */

// Standard java packages.
import java.util.Vector;

// GridStat packages.
import edu.wsu.gridstat.subscriber.derivedValue.DVHolderBase;

public abstract interface HolderBaseWithDVInterface extends HolderBaseInterface
{
  /**
  * The <code>registerDV</code> method is used to register this holder to
  * decode and store the latest value of this pattern.
  * <BR>
  * @param patternHolder Where the latest valus of the dv will be stored.
  * @return Returns <code>true</code> if we registered the pattern, note the pattern might not
  * be published, when we know this we will set Constants.PATTERN_PUBLISHED_NO in the patternHolder.
  * <code>false</code> is returned if we know that this pattern is not published or type mismatch.
  */
  public boolean registerDV(DVHolderBase patternHolder);

  /**
  * The <code>unregisterDV</code> method is used to unregister this holder from
  * decode and store the latest value of this pattern.
  * <BR>
  * @param patternId The is of the pattern to listen to.
  * @return Returns the holder that was used for this pattern.
  */
  public DVHolderBase unregisterDV(int patternId);

  /**
  * The <code>getPubDVs</code> method is used to return all the patterns that the publisher is publishing.
  * <BR>
  * @return Returns a Vector containing PatternInfoHolder of the patterns that the publisher is publishing.
  */
  public Vector getPubDVs();

  /**
  * The <code>clone</code> method is used to return a deep copy of this object
  * <BR>
  * @return Returns a deep copy of this object.
  */
  public abstract Object clone();
}
```