

```

/**
 * The <code>getSubInterval</code> method is used to return the subscription interval of this variable.
 * <BR>
 * @return Returns the subscription interval of the variable.
 */
public int[] getSubInterval();

/**
 * The <code>getExtrapolated</code> method is used to return if the current value of the variable was extrapolated or not.
 * <BR>
 * @return Returns if the current value was extrapolated or not.
 */
public boolean getExtrapolated();

/**
 * The <code>getErrorCode</code> method is used to return the current error code if any.
 * <BR>
 * @return Returns the current error code.
 */
public short getErrorCode();

/**
 * The <code>getAndResetErrorCode</code> method is used to return the current error code if any, and reset it.
 * <BR>
 * @return Returns the current error code.
 */
public short getAndResetErrorCode();

/**
 * The <code>setErrorCode</code> method is used to set the error code.
 * <BR>
 * @param error The new error code.
 */
public void setErrorCode(short error);

/**
 * The <code>registerQoSViolationListener</code> method is used to register that callbacks will be made if QoS is violated.
 * <BR>
 * @param qvi If QoS is violated this this method is called.
 * @return Returns <code>true</code> if the qvi is registered, <code>false</code> otherwise.
 */
public boolean registerQoSViolationListener(QoSViolationInterface qvi);

/**
 * The <code>isQoSViolationListener</code> method is used to return if QoSViolationListener is activated.
 * <BR>
 * @return Returns if QoSViolationListener is activated.
 */
public boolean isQoSViolationListener();

/**
 * The <code>unregisterQoSViolationListener</code> method is used to unregister the callback.
 */
public void unregisterQoSViolationListener();

/**
 * The <code>registerEventNotificationListener</code> method is used to register that the user wants a push interaction style.
 * A listener is registered that is called when an event is received.
 * <BR>
 * @param eni The listener that the user registers.
 * @return Returns <code>true</code> if the eni is registered, <code>false</code> otherwise.
 */
public boolean registerEventNotificationListener(EventNotificationInterface eni);

/**
 * The <code>isEventNotificationListener</code> method is used to find out if this holder uses push interaction.
 * <BR>
 * @return Returns <code>true</code> if push interaction is used, <code>false</code> otherwise.
 */
public boolean isEventNotificationListener();

/**
 * The <code>unregisterEventNotificationListener</code> method is used to unregister the notification of received events.
 */
public void unregisterEventNotificationListener();

/**
 * 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();
}

```