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

/**
 * <p>Title: SubscriberWithDVInterface </p>
 * <p>Description: </p>
 * <p>Copyright: Copyright (c) 2002</p>
 * <p>Company: Washington State University </p>
 * @author Kjell Harald Gjermundrod
 * @version 1.0
 */

public interface SubscriberWithDVInterface extends SubscriberInterface
{
  /**
  * The <code>subscribeInt</code> method is used to subscribe to a status variable of type int. The leaf QoS broker will set up an
  * path so that we will receive the latest value of the variable.
  * <BR>
  * @param publisherName The name of the publisher.
  * @param variableName The name of the variable to be subscribed to.
  * @param modes The modes that this subscription will be valid in.
  * @param interval The interval that we wish to receive the events.
  * @param priority The priority of the subscription.
  * @param latency The latency that we would like for this subscription.
  * @param redundancy The redundancy that we would like for this subscription.
  * @param holderObject Where the subscribed to value will be placed.
  * @return Returns true if message is subscribed to, false otherwise.
  */
  public short subscribeInt(String publisherName, String variableName, int modes, int[] interval, short[] priority,
                            short[] latency, short[] redundancy, HolderIntWithDVInterface holderObject);

  /**
  * The <code>subscribeFloat</code> method is used to subscribe to a status variable of type float. The leaf QoS broker will set up an path
  * so that we will receive the latest value of the variable.
  * <BR>
  * @param publisherName The name of the publisher.
  * @param variableName The name of the variable to be subscribed to.
  * @param modes The modes that this subscription will be valid in.
  * @param interval The interval that we wish to receive the events.
  * @param priority The priority of the subscription.
  * @param latency The latency that we would like for this subscription.
  * @param redundancy The redundancy that we would like for this subscription.
  * @param holderObject Where the subscribed to value will be placed.
  * @return Returns true if message is subscribed to, false otherwise.
  */
  public short subscribeFloat(String publisherName, String variableName, int modes, int[] interval, short[] priority,
                              short[] latency, short[] redundancy, HolderFloatWithDVInterface holderObject);

  /**
  * The <code>subscribeBoolean</code> method is used to subscribe to a status variable of type boolean. The leaf QoS broker will set up
  * an path so that we will receive the latest value of the variable.
  * <BR>
  * @param publisherName The name of the publisher.
  * @param variableName The name of the variable to be subscribed to.
  * @param modes The modes that this subscription will be valid in.
  * @param interval The interval that we wish to receive the events.
  * @param priority The priority of the subscription.
  * @param latency The latency that we would like for this subscription.
  * @param redundancy The redundancy that we would like for this subscription.
  * @param holderObject Where the subscribed to value will be placed.
  * @return Returns true if message is subscribed to, false otherwise.
  */
  public short subscribeBoolean(String publisherName, String variableName, int modes, int[] interval, short[] priority,
                                short[] latency, short[] redundancy, HolderBooleanWithDVInterface holderObject);

  /**
  * The <code>subscribeUserDefined</code> method is used to subscribe to a status variable of type boolean. The leaf QoS broker will set
  * up an path so that we will receive the latest value of the variable.
  * <BR>
  * @param publisherName The name of the publisher.
  * @param variableName The name of the variable to be subscribed to.
  * @param modes The modes that this subscription will be valid in.
  * @param interval The interval that we wish to receive the events.
  * @param priority The priority of the subscription.
  * @param latency The latency that we would like for this subscription.
  * @param redundancy The redundancy that we would like for this subscription.
  * @param holderObject Where the subscribed to value will be placed.
  * @param userType The type of the user registered type.
  * @return Returns true if message is subscribed to, false otherwise.
  */
  public short subscribeUserDefined(String publisherName, String variableName, int modes, int[] interval, short[] priority,
                                    short[] latency, short[] redundancy, HolderUserDefinedWithDVInterface holderObject, int userType);
```