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

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

// Standard java packages.
import java.util.concurrent.ArrayBlockingQueue;

// GridStat packages.
import edu.wsu.gridstat.util.IntHashMap;

public interface SubscriberInterface
{
  /**
  * The <code>connectToEdge</code> method is used to connect to a edgeStatusRouter.
  * <BR>
  * @return Returns <code>0</code> if connected, error code otherwise.
  */
  public short connectToEdge();

  /**
  * The <code>disConnectFromEdge</code> method is used to disconnect from a edgeStatusRouter.
  * <BR>
  * @return Returns <code>0</code> all subscriptions is unregistered and we disconnect
  * from the edge, error code otherwise.
  */
  public short disConnectFromEdge();

  /**
  * The <code>isConnected</code> method is used to return if we are connected to the edgeStatusRouter.
  * <BR>
  * @return Returns <code>true</code> if we are connected,
  * <code>false</code> otherwise.
  */
  public boolean isConnected();

  /**
  * The <code>getCurrentMode</code> method is used to find out the current operating mode.
  * <BR>
  * @return Return the current operating mode.
  */
  public int getCurrentMode();

  /**
  * The <code>isModeChange</code> method is used to return to the application layer if a mode change is in progress.
  * <BR>
  * @return Return <code>true</code> if a mode change is in progress, <code>false</code> otherwise.
  */
  public boolean isModeChange();

  /**
  * The <code>getNextMode</code> method is used to return the next mode if a mode change is in progress.
  * <BR>
  * @return Return the next mode, or if no mode change returns the current mode.
  */
  public int getNextMode();

  /**
  * 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, HolderIntInterface holderObject);
```