`JavaRosWrapper`: Located in `Cyphyhouse/ros`

`JavaRosWrapper` serves as the wrapper which allows Java code to publish and subscribe to ROS topics. This object is currently instantiated in various `MotionAutomaton` files to allow the different platforms to:

1. Send destination messages (of type: ros.msgs.geometry_msgs.Point)
2. Receive ack messages from hardware controller acknowledging that a destination has been reached.

General Usage Guideline:
In order to properly use the `JavaRosWrapper` to communicate with ROS, one must:

1. Instantiate `JavaRosWrapper` object and provide it with a port string (set to ws://localhost:9090 if using default rosbridge_server package).
2. Use the `createTopic` method to create any new ROS topics which Java will be publishing to.
3. Use the `subscribe_to_ROS` method to subscribe to any ROS topics for which the Java code should get new messages from.
4. Use the `sendMsg` method to publish messages to a previously created ROS topic.

Supported Message Types:
The current `JavaRosWrapper` version supports a number of common ROS message types. All supported message types can be found in the `RosJavaBridge` GitHub repository. Additional message types can be added there as needed in the future. After adding the new message types, compile the project in `RosJavaBridge` and add the .jar files in the `target` directory to the `targets` directory in `newLib`.

Callback Functions for Subscribed Topics:
The `subscribe_to_ROS` method of the `JavaRosWrapper` class contains a list of all the currently supported topics. In order to add support for subscription to a new topic with a specific callback function, append to the case statement with the appropriate topic. Note that you must define a callback function for new topic subscriptions.

High-level Layout: