Code Snippets – IntelliWare 1.0

```java
public class GenericDevice {
    protected ArrayList<Function> functions = new ArrayList<Function>();
    public boolean doFunction(String function, String value) {
       // loop through device functions to find desired function
       for (int i = 0; i < functions.size(); i++) {
              // verify if we have match
                     if (functions.get(i).getName().equals(function)) {
                            // create boolean to store result of function execution
                            boolean bResult;

                            // execute function, with or without parameter depending on input
                            if (value.isEmpty()) {
                                   bResult = functions.get(i).doFunction();
                            } else {
                                   bResult = functions.get(i).doFunction(value);
                            }
                            // display function result (success or failure)
                            if (bResult) {
                                   // print what executed function
                                   System.out.println(
                                   "function '" + function + value + "' peformed by " + name);
                            } else {
                                   System.out.println(
                                   "function '" + function + value +"' can not be executed by "
                                   + name + " (requires different arguments)");
                            }
                            // return we executed command
                            return true;
                     }
              }
       // function not found, not supported by device, print message
       System.out.println("function '" + function + "' not supported by " + name);

       // return false
       return false;
    }
}
```

Any functions can be passsed to any device however the Generic Device class will always filter out those requests that a device can't handle. The success or failure of a function will be returned.

```java
public interface Function {
    public String getName();
    public boolean doFunction();
    public boolean doFunction(String value);
}

public class TV extends GenericDevice {
    public TV() {
        name = "TV";
        functions.add(new OnFunction());
        functions.add(new OffFunction());
        functions.add(new SetInputChannelFunction());
        functions.add(new SetVolumeFunction());
    }
}
```

You can create new devices and send commands directly to them. You can simulate any command being send to the device and the device class will filter out the invalid commands.

```java
TV myTV = new TV();
myTV.doFunction("ON");
myTV.doFunction("SET_VOLUME", "20");
```

You could also manipulate multiple devices by using a Generic Device arraylist and a simple for each loop as illustrated below.

```java
oaAllDevices.add(myFan);
oaAllDevices.add(myAlarm);
oaAllDevices.add(myTV);
for (GenericDevice device : oaAllDevices) {
       device.doFunction("OFF");
}
```

```java
public class OnFunction implements Function {
   @Override
   public String getName() {
          return "ON";
   }
   @Override
       public boolean doFunction() {
          System.out.println("Turn On");
          return true;
       }
   @Override
   public boolean doFunction(String value) {
          return doFunction();
   }
}
```