Auto-generating Javadoc comment style using Netbeans

Right click the Java file. Select ‘Tools’ and then ‘Analyze Javadocs’. (Screenshot 1)

The tool will let you know all the methods and classes that have not been documented in the Javadocs style. Select all of those and click the button ‘Fix Selected’. (Screenshot 2)

After this, follow the guidelines below to document your code well.
/**
 * Author: Michael McCarthy
 * Last Modified: May 27, 2009
 *
 * This program demonstrates a very simple UDP client.
 * The command line contains two operands and an operator.
 * The command line is packaged and placed in a single UDP packet.
 * The packet is sent to the server asynchronously.
 * The program then blocks waiting for the server to perform
 * the requested operation. When the response packet arrives,
 * a String object is created and the reply is displayed.
 * The program illustrates the marshaling and un-marshaling
 * of requests and replies. It also demonstrates that UDP
 * style sockets are quite different from TCP style sockets.
 */

// imports required for UDP/IP
import java.net.*;
import java.io.*;

public class UDPClient {

    /**
     * The doFlickrSearch is used to get the Image URL for the corresponding
     * animal
     * @param strAnimalName The name of the animal
     * @return picURL The URL of the Image for the corresponding animal
     * @throws Exception  The Exception thrown is any malfunction occurs
     */
    public String doFlickrSearch(String strAnimalName) throws Exception {
        String returnString = null;
        String[] arrString = new String[2];
        try {
            // Create a URL for the desired animal
            URL url = new URL("http://www.flickr.com/search/?s=int&q=" + strAnimalName);
            // Read all the text returned by the Flickr server
            BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));
            String strLine;
            while ((strLine = in.readLine()) != null) {
                returnString += strLine;
            }
        }
    }
}
// If image could not be found throw an exception to the calling class
// by passing a message "Did not find a Image for the alphabet"
if (null == returnString || returnString.length() == 0) {
    throw new Exception("Did not find a Image for the alphabet");
}

in.close();
} catch (MalformedURLException e) {
    throw new Exception("Did not find a Image for the alphabet " +
    e.getMessage());
}
} catch (Exception e) {
    throw new Exception("Did not find a Image for the alphabet " +
    e.getMessage());
}
}