```
import java.util.HashMap;
import java.util.Map;

class ARPCache {
    private Map<String, String> cache;

    public ARPCache() {
        this.cache = new HashMap<>();
    }

    public void addToCache(String ipAddress, String macAddress) {
        cache.put(ipAddress, macAddress);
    }

    public String getMacAddress(String ipAddress) {
        return cache.get(ipAddress);
    }
}

class ARPSimulator {
    private ARPCache arpCache;

    public ARPSimulator(ARPCache arpCache) {
        this.arpCache = arpCache;
    }

    public void simulateARP(String ipAddress) {
        String macAddress = getMacAddressFromARP(ipAddress);
        System.out.println("MAC Address for IP " + ipAddress + ": " + macAddress);
    }

    private String getMacAddressFromARP(String ipAddress) {
        String macAddress = arpCache.getMacAddress(ipAddress);

        if (macAddress == null) {
            macAddress = generateRandomMAC();
            arpCache.addToCache(ipAddress, macAddress);
        }
        return macAddress;
    }

    private String generateRandomMAC() {
        return "00:1A:2B:3C:4D:5E";
    }
}

public class ARPSimulation {
    public static void main(String[] args) {
        ARPCache arpCache = new ARPCache();
        ARPSimulator arpSimulator = new ARPSimulator(arpCache);

        arpSimulator.simulateARP("192.168.1.1");
        arpSimulator.simulateARP("192.168.1.2");
        arpSimulator.simulateARP("192.168.1.1");
    }
}
```

```
import java.util.HashMap;
import java.util.Map;

class RARPCache {
    private Map<String, String> cache;

    public RARPCache() {
        this.cache = new HashMap<>();
    }

    public void addToCache(String macAddress, String ipAddress) {
        cache.put(macAddress, ipAddress);
    }

    public String getIPAddress(String macAddress) {
        return cache.get(macAddress);
    }
}

class RARPSimulator {
    private RARPCache rarpCache;

    public RARPSimulator(RARPCache rarpCache) {
        this.rarpCache = rarpCache;
    }

    public void simulateRARP(String macAddress) {
        String ipAddress = getIPAddressFromRARP(macAddress);
        System.out.println("IP Address for MAC " + macAddress + ": " + ipAddress);
    }

    private String getIPAddressFromRARP(String macAddress) {
        String ipAddress = rarpCache.getIPAddress(macAddress);

        if (ipAddress == null) {
            ipAddress = generateRandomIP();
            rarpCache.addToCache(macAddress, ipAddress);
        }
        return ipAddress;
    }

    private String generateRandomIP() {
        return "192.168.1.1";
    }
}

public class RARPSimulation {
    public static void main(String[] args) {
        RARPCache rarpCache = new RARPCache();
        RARPSimulator rarpSimulator = new RARPSimulator(rarpCache);

        rarpSimulator.simulateRARP("00:1A:2B:3C:4D:5E");
        rarpSimulator.simulateRARP("00:1A:2B:3C:4D:5F");
        rarpSimulator.simulateRARP("00:1A:2B:3C:4D:5E");
    }
}
```