

```

                                VendCli
// VendCli.java - VENDING MACHINE CLIENT SIMULATOR
//
// USAGE
// java VendCli [ c | e ]
//
// c          indicates that the client simulator is to emulate the
//             coin acceptor
// e          indicates that the client simulator is to emulate the
//             e-purse interface
//
// MODULE INDEX
// NAME                                CONTENTS
// VendCli                             Vending machine client class
// VendCli.VendCli                     Construct a vending machine client
// VendCli.main                         Main line
//
// MAINTENANCE HISTORY
// DATE          PROGRAMMER AND DETAILS
// 03-09-05     JS          Original
// 05-09-05     LSH        Fixed crash in arguement check
// 05-09-05     LSH        Modified to resend request when connection failed
//
//-----
// IMPORTATIONS
import java.io.*;
import java.util.*;
import java.net.*;

//-----
// VENDING MACHINE CLIENT CLASS
class VendCli
{
    //-----

    // CONSTRUCT A VENDING MACHINE CLIENT
    VendCli (
        int          clientPort)    // client port number
    {
        Socket       comSocket;    // Socket to communicate w/ client
        InputStream  comInputStream; // Communications input stream
        OutputStream comOutputStream; // Communications output stream
        int          ofs;          // Offset in receive buffer
        int          rcvdLen;      // Received length
        byte []      respBuf;      // Response buffer
        VendMux.Unpack  unpack;    // Unpacker reference
        VendMux.Pack    pack;      // Packer reference
        long         serialNo;     // Request serial number
        long         available;    // Available funds
        long         consumed;     // Consumed amount
        Random       rand;         // A random number generator
        boolean      resendReq;    // Resend request flag

        // Instantiate a random number generator
        rand = new Random ();

        // Select an initial serial number
        serialNo = rand.nextLong() & 0xffffffffL;

        // Initialise the available fund amount (make lint happy)
        available = 0;

        // Set the resend request flag
        resendReq = false;

        // Loop through connection attempts
        for (;;) {
            // Initialise the socket references

```

```

                                VendCli
comSocket = null;
comInputStream = null;
comOutputStream = null;

// Catch I/O exceptions
try {
    // Open a socket to connect to the server
    comSocket = new Socket (
        InetAddress.getBy_name(VendMux.VEND_NAME),
        VendMux.MUX_LISTEN_PORT,
        InetAddress.getLocalHost(),
        clientPort
    );
    comInputStream = comSocket.getInputStream();
    comOutputStream = comSocket.getOutputStream();

    // Loop to process requests
    for (;;) {
        // Generate a new request if the previous request is successful
        if (resendReq == false) {
            // Select the next serial number
            serialNo = (serialNo + 1) & 0xffffffffL;

            // Select an available amount
            available = rand.nextInt (1000) + 1;
        }

        // Encode the request
        pack = new VendMux.Pack ( VendMux.REQUEST_LEN );
        pack.packNumber (serialNo, 4);
        pack.packNumber (available, 4);

        // Send the request
        comOutputStream.write (pack.packBuf, 0, pack.packLen);

        // Receive a response from the server
        respBuf = new byte [ VendMux.RESPONSE_LEN ];
        ofs = 0;
        while (ofs < VendMux.RESPONSE_LEN) {
            rcvdLen = comInputStream.read (
                respBuf, ofs, VendMux.RESPONSE_LEN-ofs
            );
            if (rcvdLen == -1)
                throw new IOException ("end-of-file");
            ofs += rcvdLen;
        }

        // Unpack the response
        unpack = new VendMux.Unpack (respBuf);
        serialNo = unpack.unpackNumber (4);
        consumed = unpack.unpackNumber (4);

        // Mark request attempt as successful
        resendReq = false;

        // Display the available and consumed amounts
        System.out.println ("VendCli: available=" + available
            + " consumed=" + consumed);

        // Wait 1/2 second before making another request
        try {
            Thread.currentThread().sleep (500);
        }
        catch (InterruptedException e2) {

```

```

                                VendCli
                                // Empty
                                }
                                }
                                }
                                catch (IOException e) {
                                    System.err.println ("warning: client: "
                                        + e.toString());

                                    // Enable the resend request flag

                                    resendReq = true;
                                }

                                // Close the socket
                                VendMux.muxCloseSocket (comSocket, comInputStream, comOutputStream);

                                // wait for 10 seconds before trying again

                                try {
                                    Thread.currentThread().sleep (10000);
                                }
                                catch (InterruptedException e2) {
                                    // Empty
                                }
                                }
                                }

                                //-----
                                // MAIN LINE
                                static public void
                                main (
                                    String []      argv)          // Argument values
                                {
                                    if (argv.length != 1) {
                                        System.err.println ("Usage: java vendCli [ c | e ]");
                                        System.exit (1);
                                    }

                                    if (argv[0].equals("c"))
                                        new VendCli (VendMux.COIN_PORT);
                                    else if (argv[0].equals("e"))
                                        new VendCli (VendMux.E_PURSE_PORT);
                                    else {
                                        System.err.println ("Usage: java vendCli [ c | e ]");
                                        System.exit (1);
                                    }
                                }
                                }
}

```