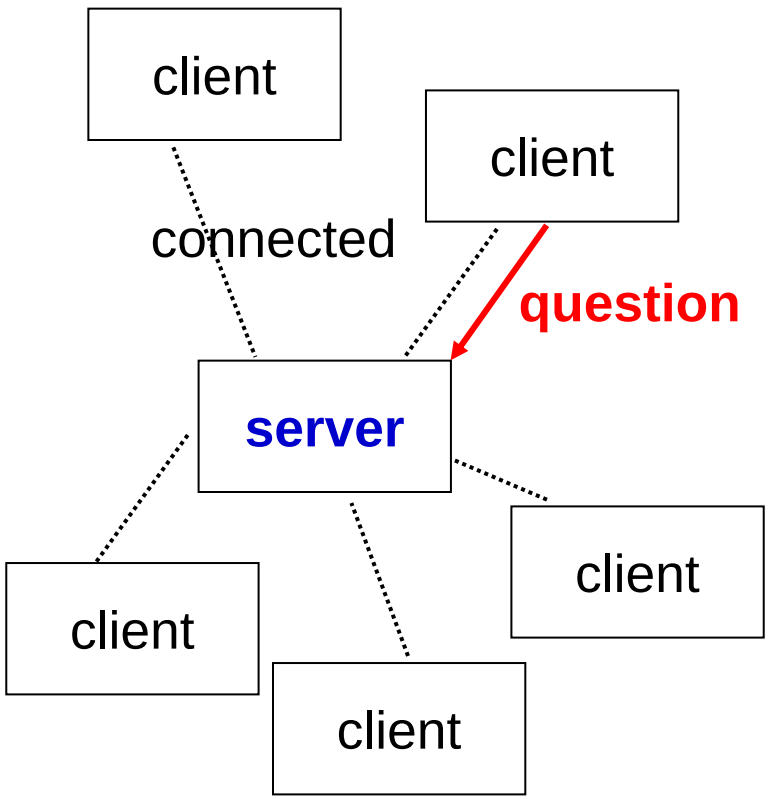
ECE 462 C++ and Java

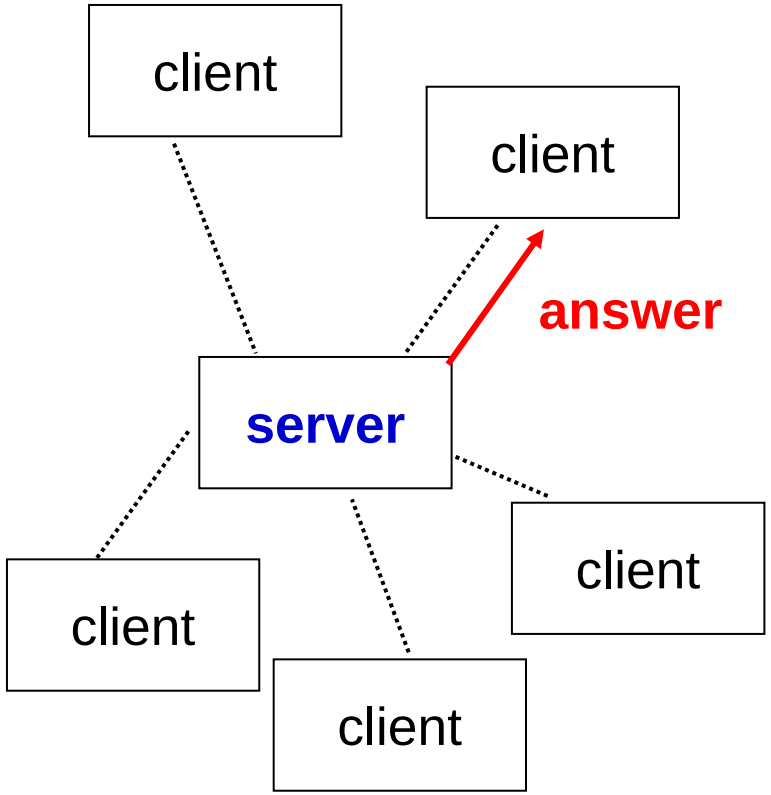
Lab Exercise 09
Networking using
Java Server and C++ Client

Yung-Hsiang Lu

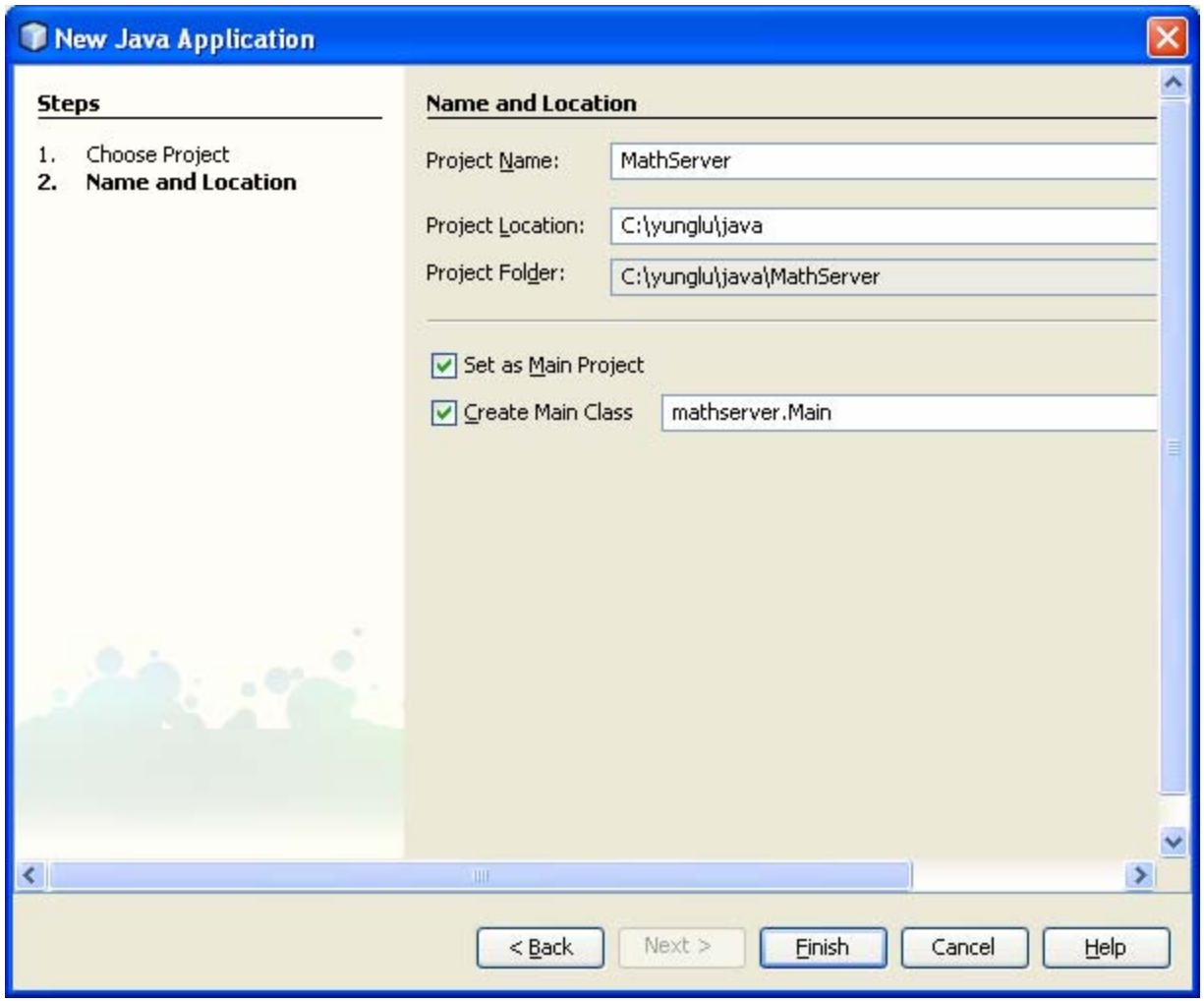
yunglu@purdue.edu

Math Server

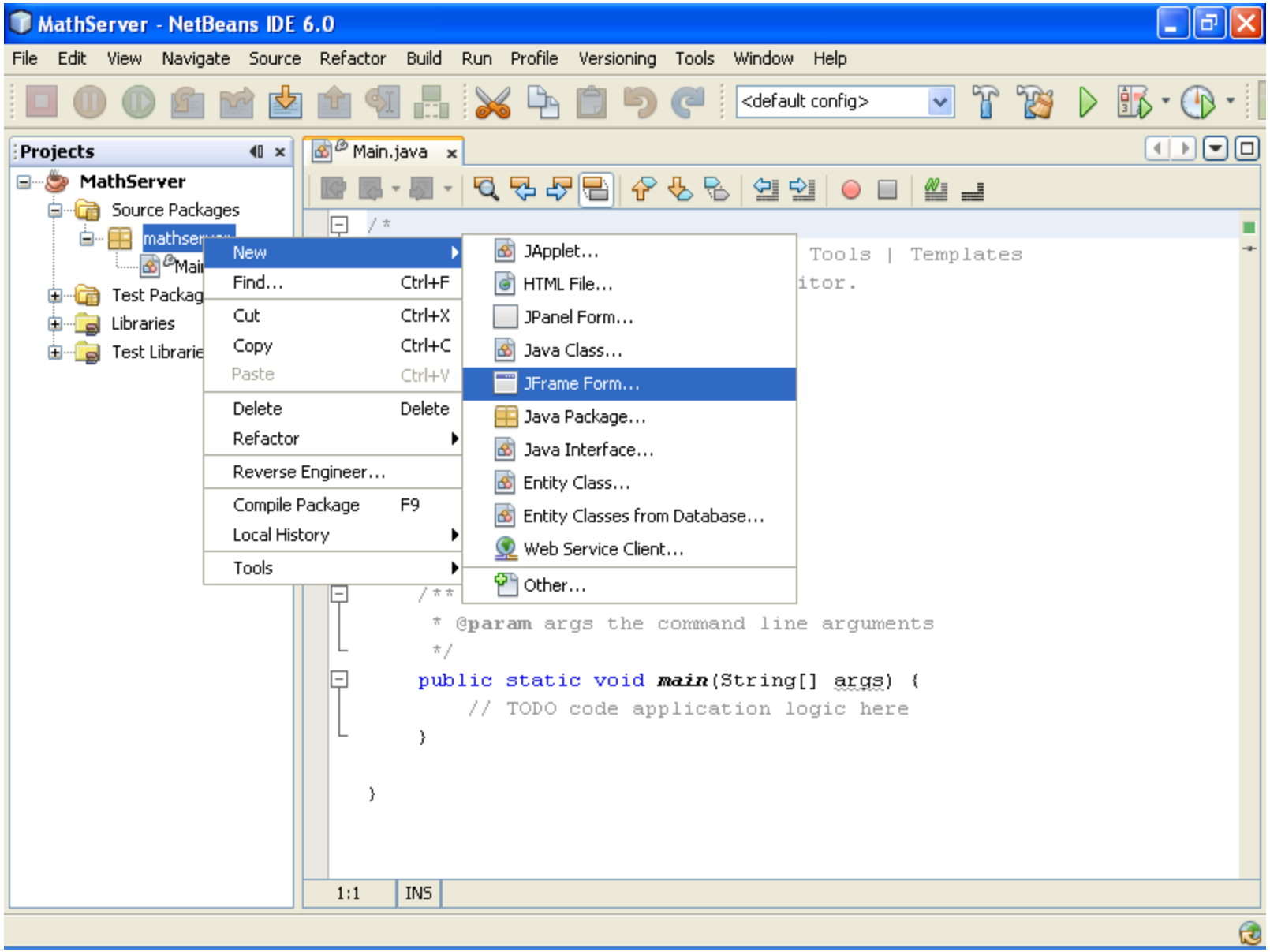

client
client
connected
question
server
client
client
client

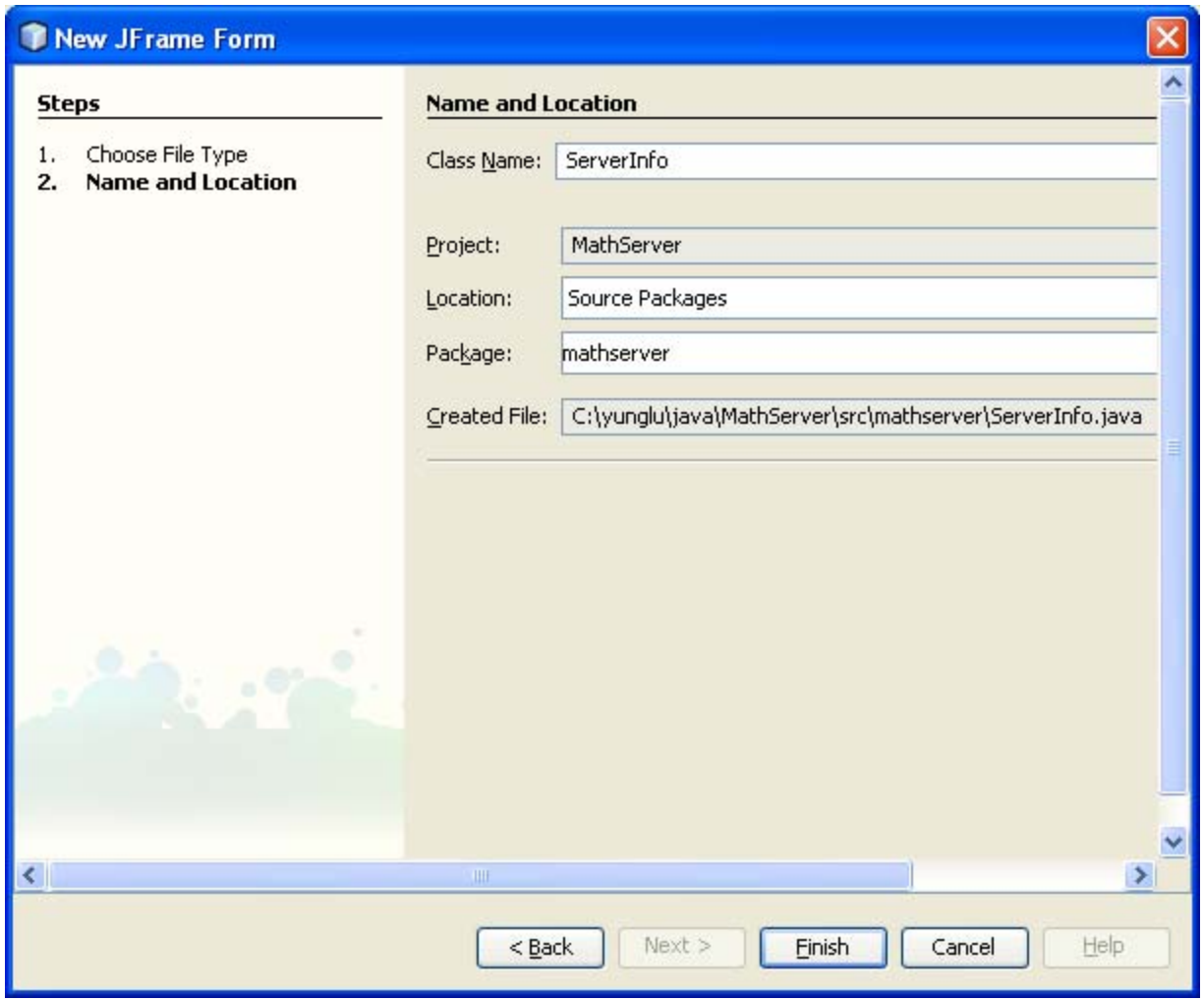

client
client
answer
server
client
client
client

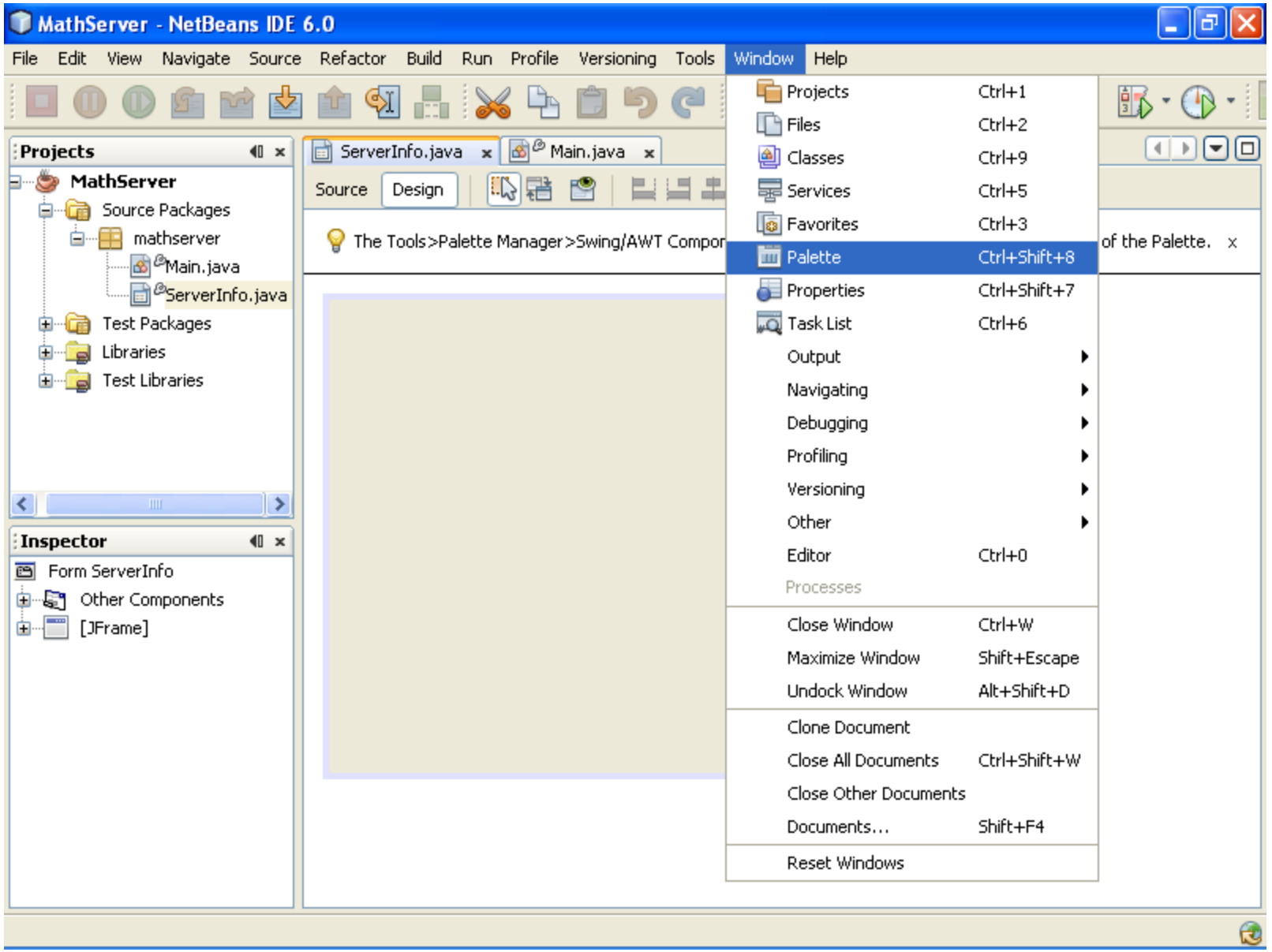
Create Java Server

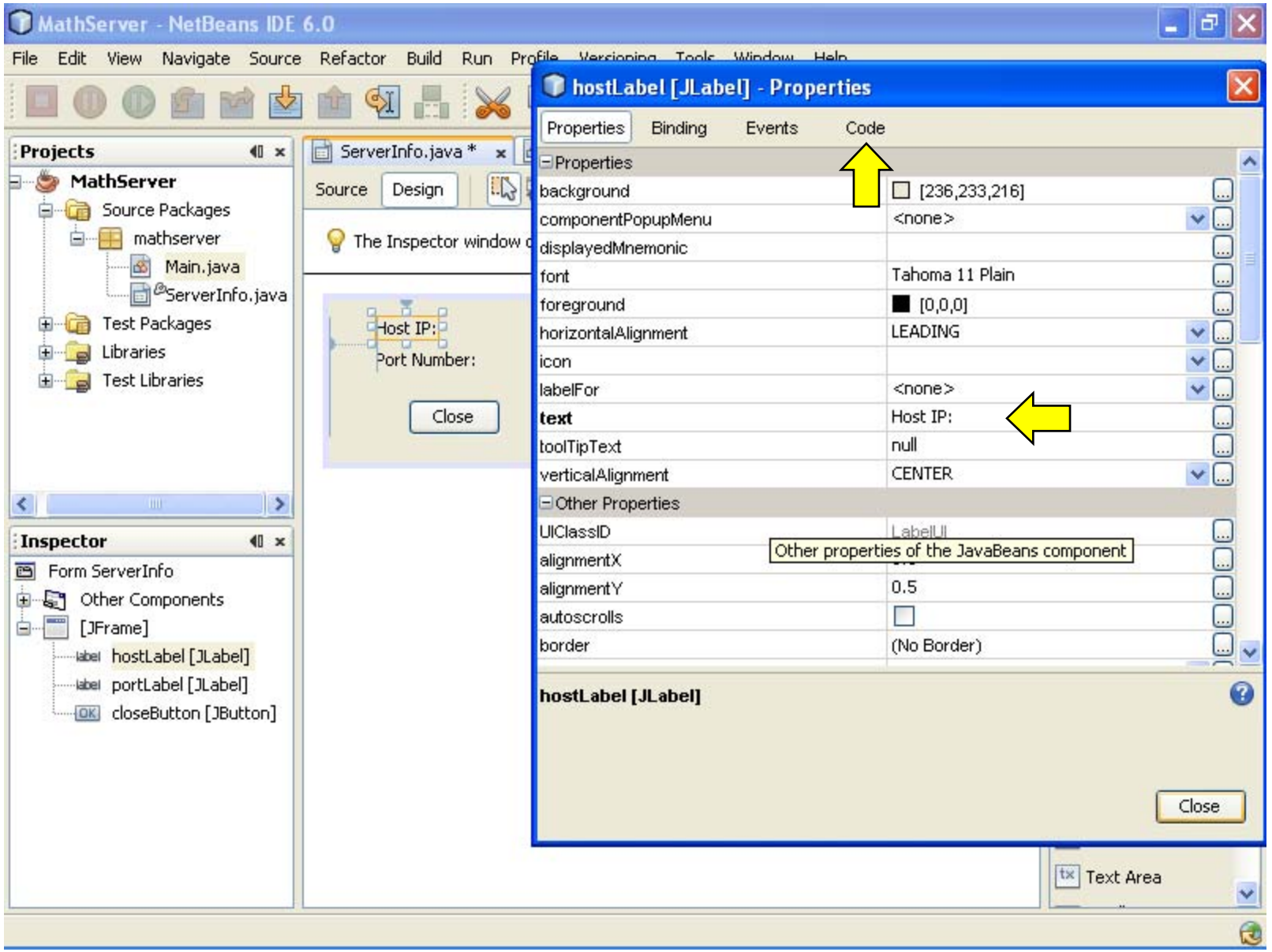

New Java Application
Steps
1. Choose Project
2. Name and Location
Name and Location
Project Name: MathServer
Project Location: C:\yunglu\java
Project Folder: C:\yunglu\java\MathServer
Set as Main Project
Create Main Class mathserver.Main
< Back
Next >
Finish
Cancel
Help

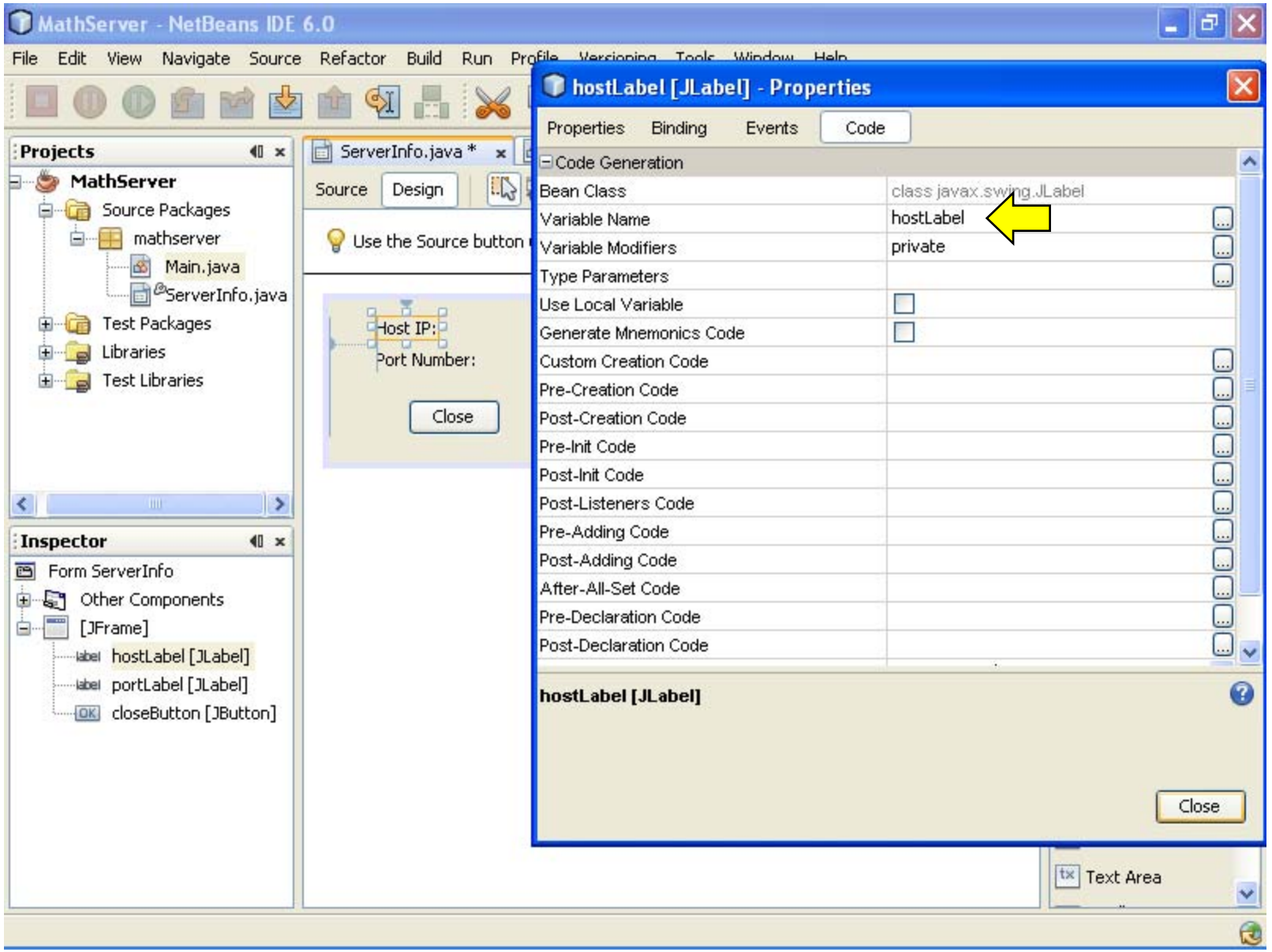
GUI for Server Information

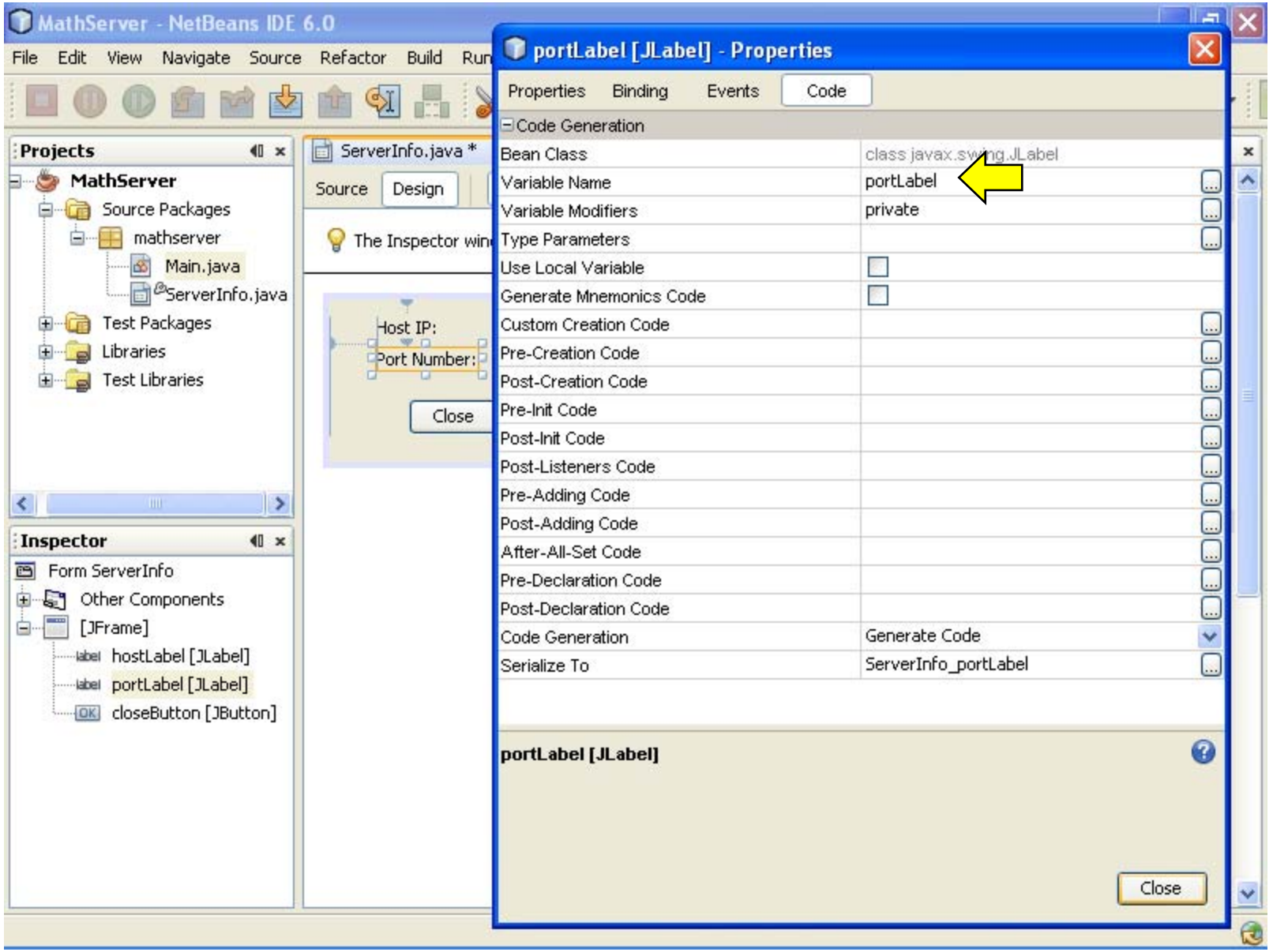

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
MathServer
Source Packages
Test Packag
Libraries
Test Librarie
Main.java
New
Find... Ctrl+F
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Refactor
Reverse Engineer...
Compile Package F9
Local History
Tools
JApplet...
HTML File...
JPanel Form...
Java Class...
JFrame Form...
Java Package...
Java Interface...
Entity Class...
Entity Classes from Database...
Web Service Client...
Other...
Tools | Templates
itor.
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
}
}
1:1
INS

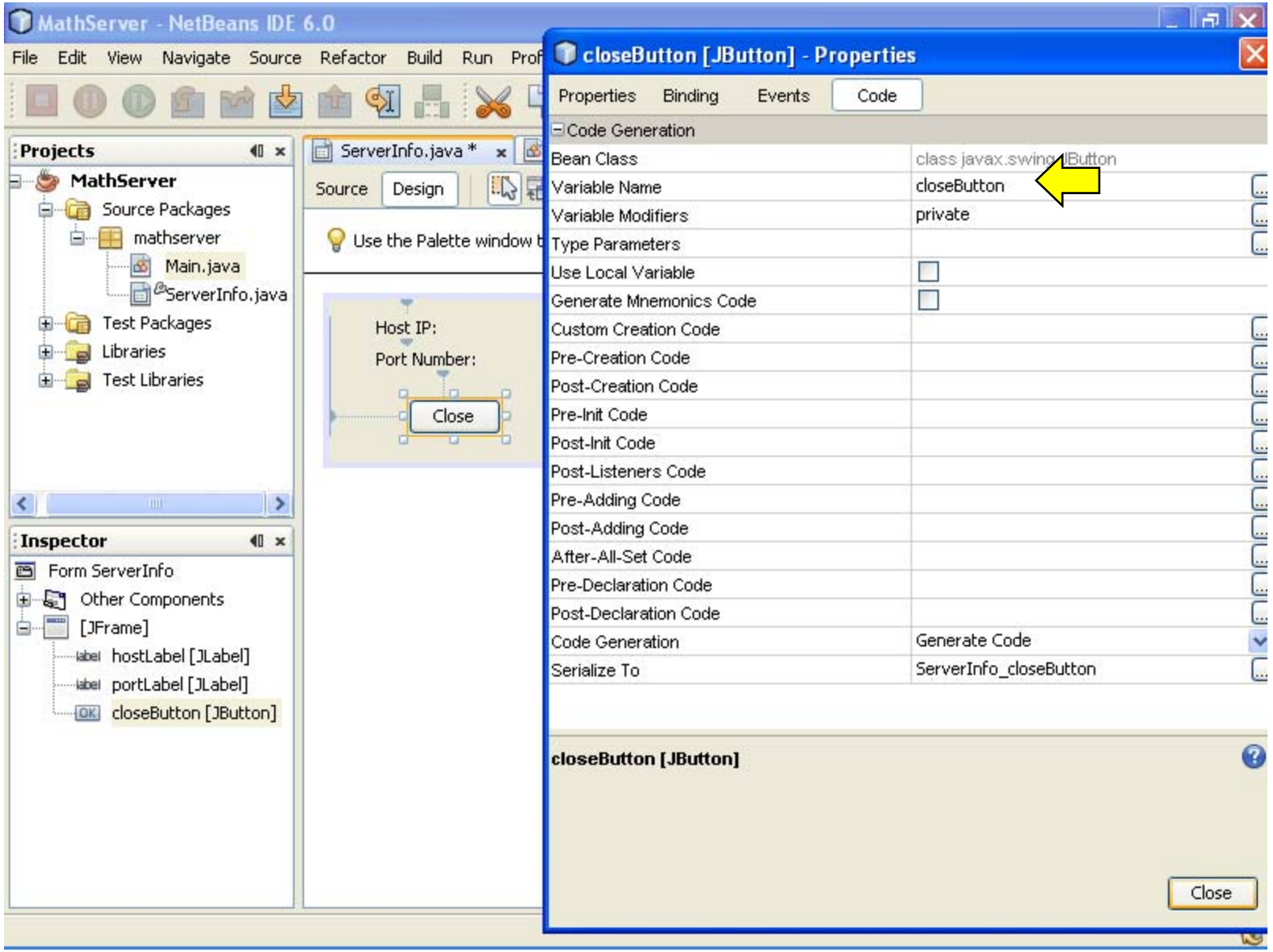

New JFrame Form
Steps
1. Choose File Type
2. Name and Location
Name and Location
Class Name: ServerInfo
Project: MathServer
Location: Source Packages
Package: mathserver
Created File: C:\yunglu\java\MathServer\src\mathserver\ServerInfo.java
< Back
Next >
Finish
Cancel
Help

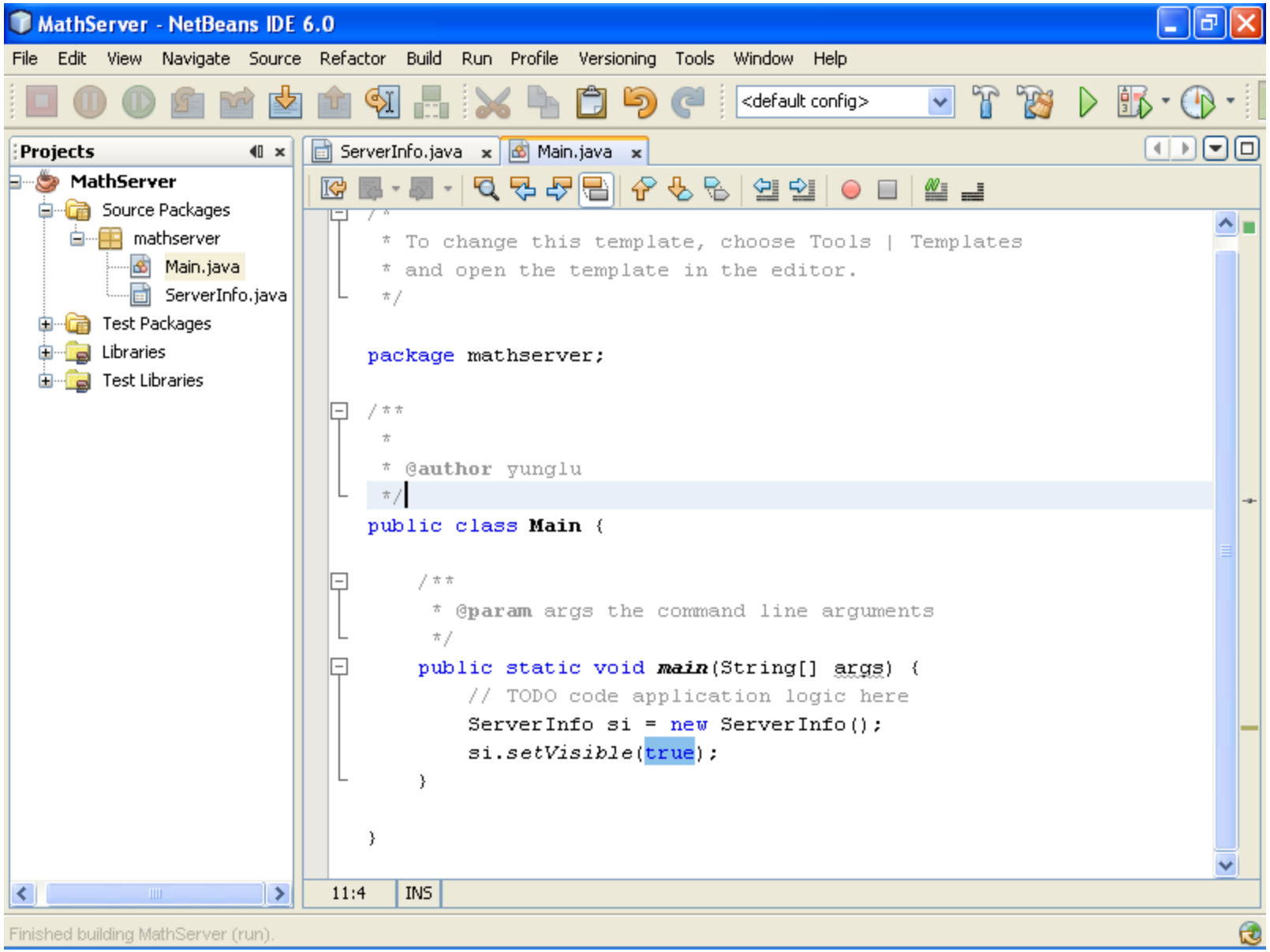

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
Inspector
Form ServerInfo
Other Components
[JFrame]
ServerInfo.java
Main.java
Source Design
The Tools>Palette Manager>Swing/AWT Compor
of the Palette.
Projects Ctrl+1
Files Ctrl+2
Classes Ctrl+9
Services Ctrl+5
Favorites Ctrl+3
Palette Ctrl+Shift+8
Properties Ctrl+Shift+7
Task List Ctrl+6
Output
Navigating
Debugging
Profiling
Versioning
Other
Editor Ctrl+0
Processes
Close Window Ctrl+W
Maximize Window Shift+Escape
Undock Window Alt+Shift+D
Clone Document
Close All Documents Ctrl+Shift+W
Close Other Documents
Documents... Shift+F4
Reset Windows

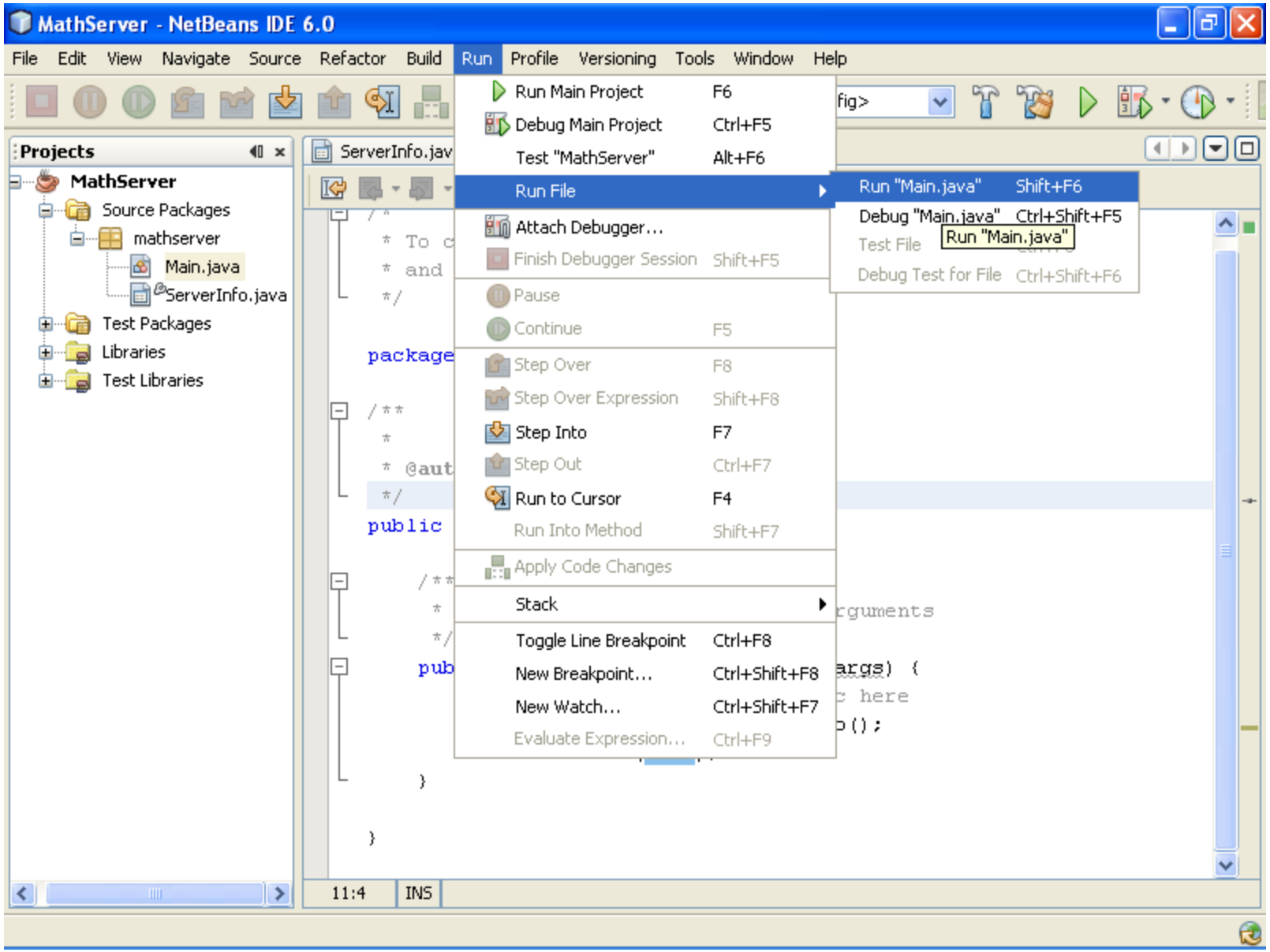

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
Inspector
Form ServerInfo
Other Components
[JFrame]
hostLabel [JLabel]
portLabel [JLabel]
closeButton [JButton]
ServerInfo.java *
Source Design
The Inspector window
Host IP:
Port Number:
Close
hostLabel [JLabel] - Properties
Properties Binding Events Code
Properties
background [236,233,216]
componentPopupMenu <none>
displayedMnemonic
font Tahoma 11 Plain
foreground [0,0,0]
horizontalAlignment LEADING
icon
labelFor <none>
text Host IP:
toolTipText null
verticalAlignment CENTER
Other Properties
UIClassID
alignmentX
alignmentY 0.5
autoscrolls
border (No Border)
Other properties of the JavaBeans component
hostLabel [JLabel]
Close
Text Area


MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
Inspector
Form ServerInfo
Other Components
[JFrame]
hostLabel [JLabel]
portLabel [JLabel]
closeButton [JButton]
ServerInfo.java *
Source Design
Use the Source button
Host IP:
Port Number:
Close
hostLabel [JLabel] - Properties
Properties Binding Events Code
Code Generation
Bean Class
class javax.swing.JLabel
Variable Name
hostLabel
Variable Modifiers
private
Type Parameters
Use Local Variable
Generate Mnemonics Code
Custom Creation Code
Pre-Creation Code
Post-Creation Code
Pre-Init Code
Post-Init Code
Post-Listeners Code
Pre-Adding Code
Post-Adding Code
After-All-Set Code
Pre-Declaration Code
Post-Declaration Code
hostLabel [JLabel]
Close
Text Area


MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
Inspector
Form ServerInfo
Other Components
[JFrame]
hostLabel [JLabel]
portLabel [JLabel]
closeButton [JButton]
ServerInfo.java *
Source
Design
The Inspector win
Host IP:
Port Number:
Close
portLabel [JLabel] - Properties
Properties
Binding
Events
Code
Code Generation
Bean Class
class javax.swing.JLabel
Variable Name
portLabel
Variable Modifiers
private
Type Parameters
Use Local Variable
Generate Mnemonics Code
Custom Creation Code
Pre-Creation Code
Post-Creation Code
Pre-Init Code
Post-Init Code
Post-Listeners Code
Pre-Adding Code
Post-Adding Code
After-All-Set Code
Pre-Declaration Code
Post-Declaration Code
Code Generation
Generate Code
Serialize To
ServerInfo_portLabel
portLabel [JLabel]
Close

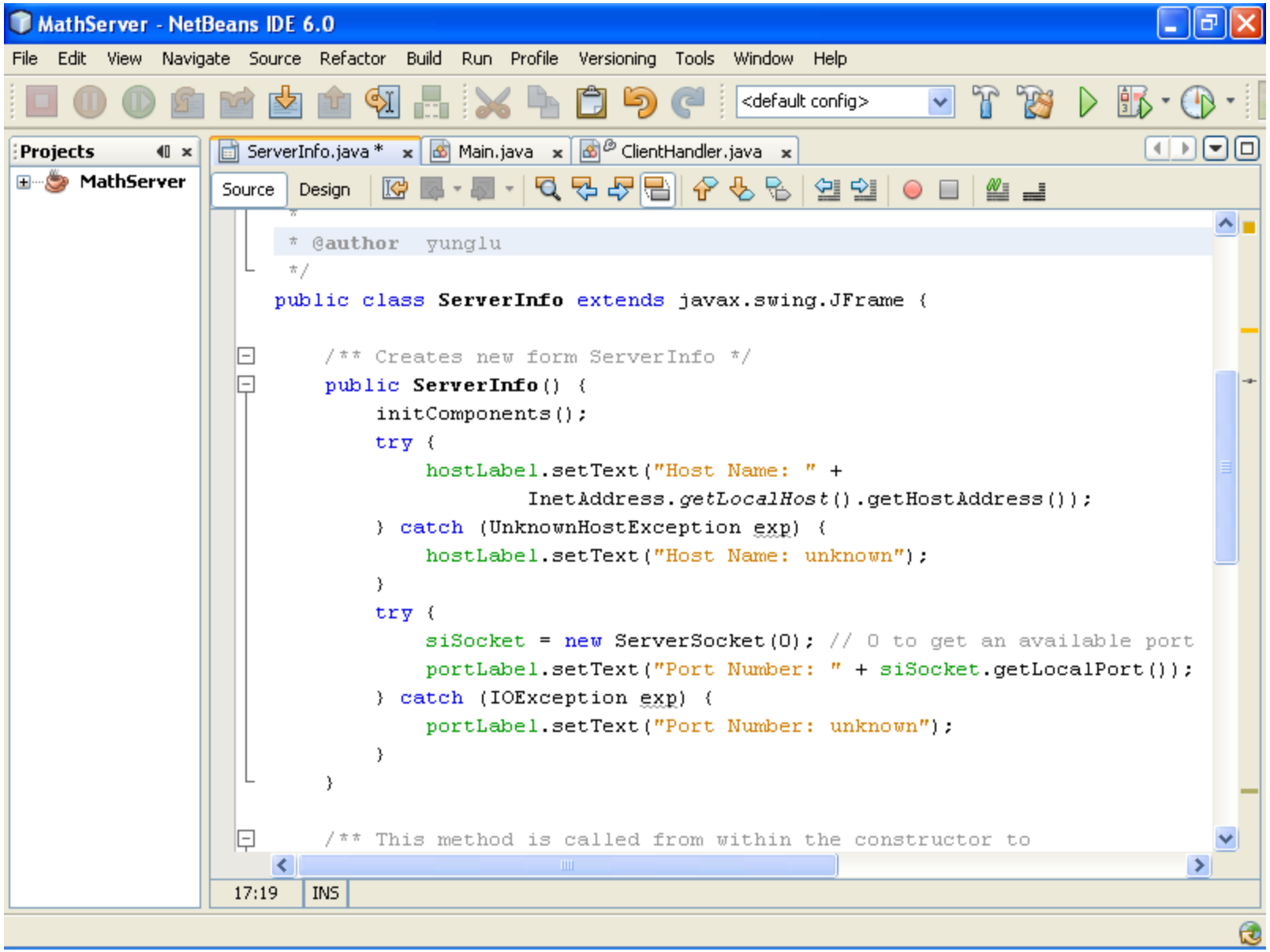

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
Inspector
Form ServerInfo
Other Components
[JFrame]
hostLabel [JLabel]
portLabel [JLabel]
closeButton [JButton]
ServerInfo.java *
Source
Design
Use the Palette window
Host IP:
Port Number:
Close
closeButton [JButton] - Properties
Properties
Binding
Events
Code
Code Generation
Bean Class
class javax.swing.JButton
Variable Name
closeButton
Variable Modifiers
private
Type Parameters
Use Local Variable
Generate Mnemonics Code
Custom Creation Code
Pre-Creation Code
Post-Creation Code
Pre-Init Code
Post-Init Code
Post-Listeners Code
Pre-Adding Code
Post-Adding Code
After-All-Set Code
Pre-Declaration Code
Post-Declaration Code
Code Generation
Generate Code
Serialize To
ServerInfo_closeButton
closeButton [JButton]
Close

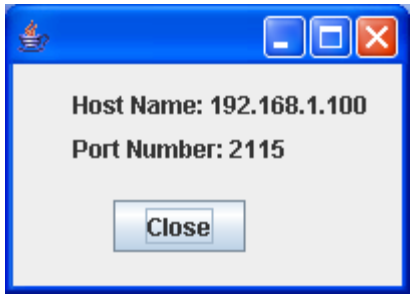

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
ServerInfo.java
Main.java
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package mathserver;
/**
*
* @author yunglu
*/
public class Main {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
ServerInfo si = new ServerInfo();
si.setVisible(true);
}
}
11:4
INS
Finished building MathServer (run).

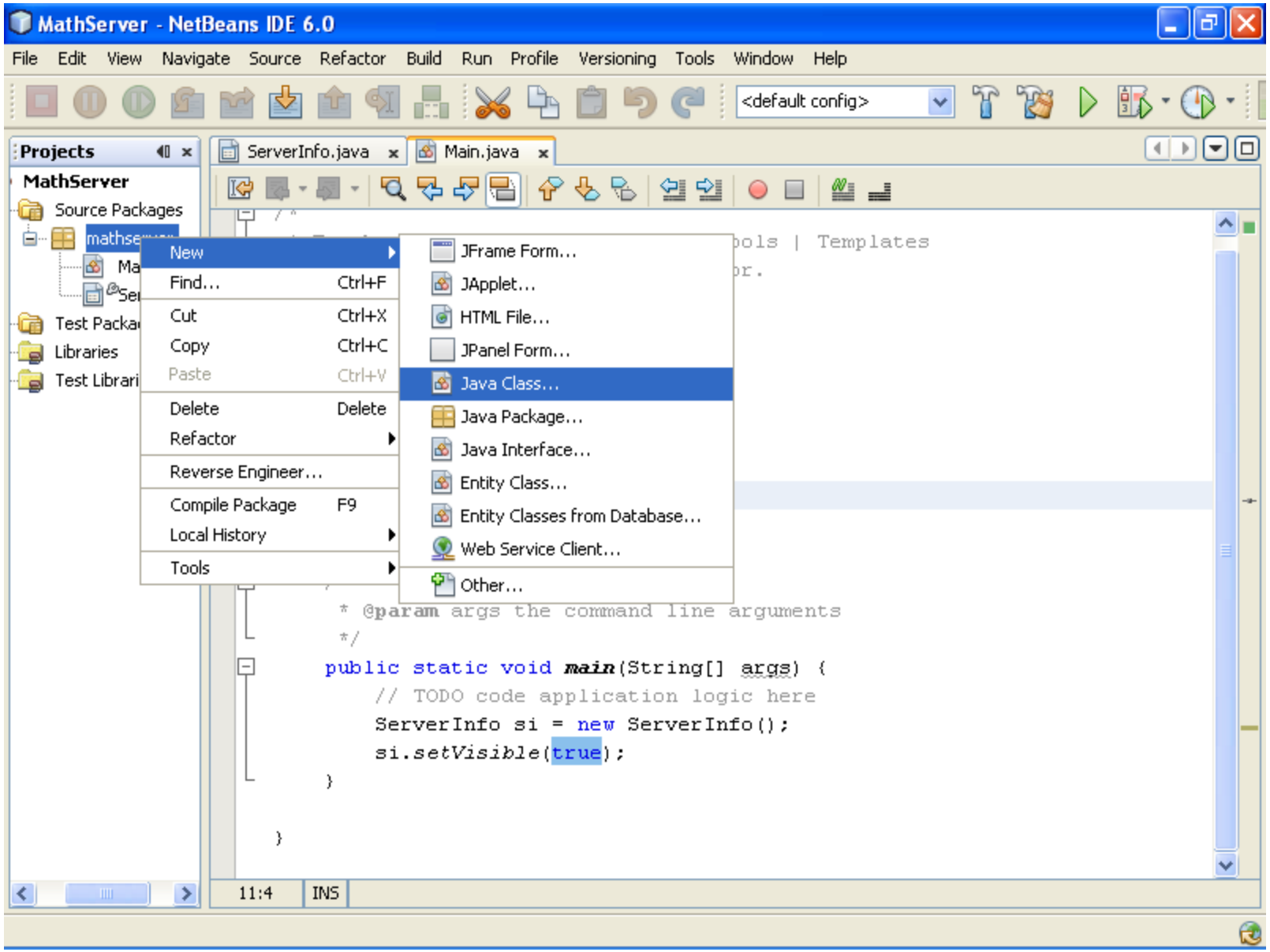

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
Run Main Project F6
Debug Main Project Ctrl+F5
Test "MathServer" Alt+F6
Run File
Attach Debugger...
Finish Debugger Session Shift+F5
Pause
Continue F5
Step Over F8
Step Over Expression Shift+F8
Step Into F7
Step Out Ctrl+F7
Run to Cursor F4
Run Into Method Shift+F7
Apply Code Changes
Stack
Toggle Line Breakpoint Ctrl+F8
New Breakpoint... Ctrl+Shift+F8
New Watch... Ctrl+Shift+F7
Evaluate Expression... Ctrl+F9
Run "Main.java" Shift+F6
Debug "Main.java" Ctrl+Shift+F5
Test File
Run "Main.java"
Debug Test for File Ctrl+Shift+F6
Projects
MathServer
Source Packages
mathserver
Main.java
ServerInfo.java
Test Packages
Libraries
Test Libraries
ServerInfo.jav
11:4
INS

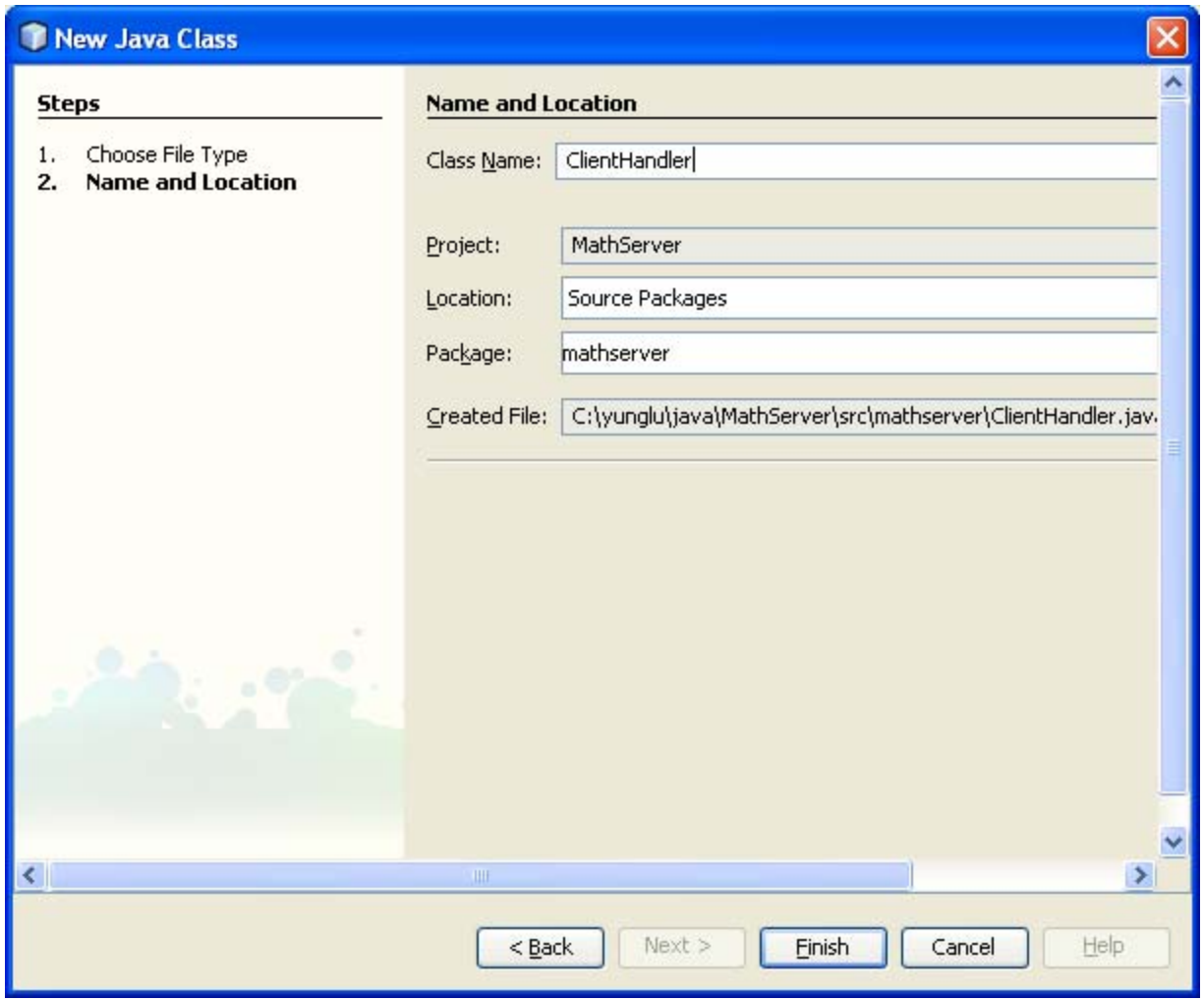

Host IP:
Port Number:
Close

IP Address and Port Number

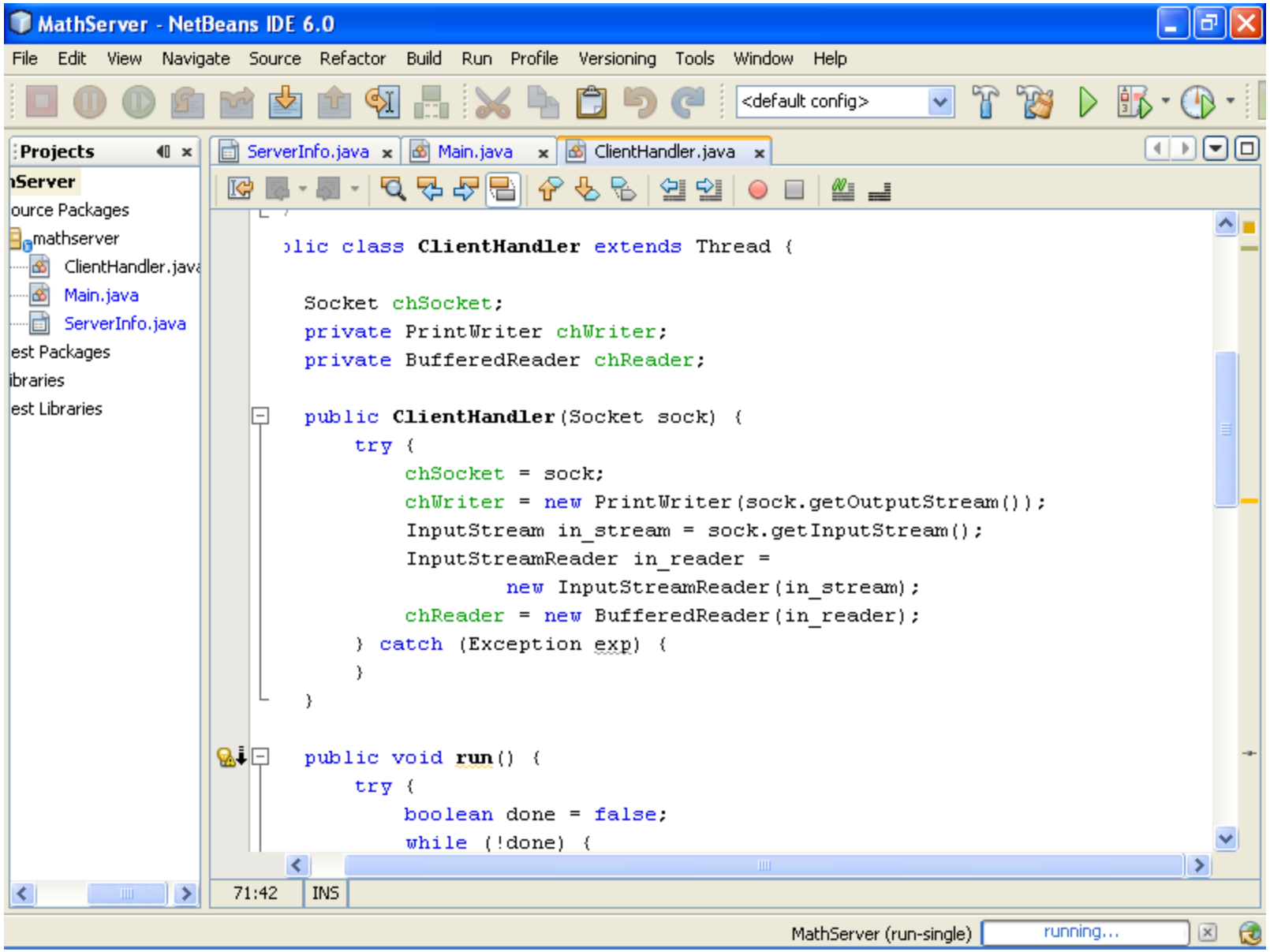

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
MathServer
ServerInfo.java *
Main.java
ClientHandler.java
Source
Design
* @author yunglu
*/
public class ServerInfo extends javax.swing.JFrame {
/** Creates new form ServerInfo */
public ServerInfo() {
initComponents();
try {
hostLabel.setText("Host Name: " +
InetAddress.getLocalHost().getHostAddress());
} catch (UnknownHostException exp) {
hostLabel.setText("Host Name: unknown");
}
try {
siSocket = new ServerSocket(0); // 0 to get an available port
portLabel.setText("Port Number: " + siSocket.getLocalPort());
} catch (IOException exp) {
portLabel.setText("Port Number: unknown");
}
}
/** This method is called from within the constructor to
17:19
INS

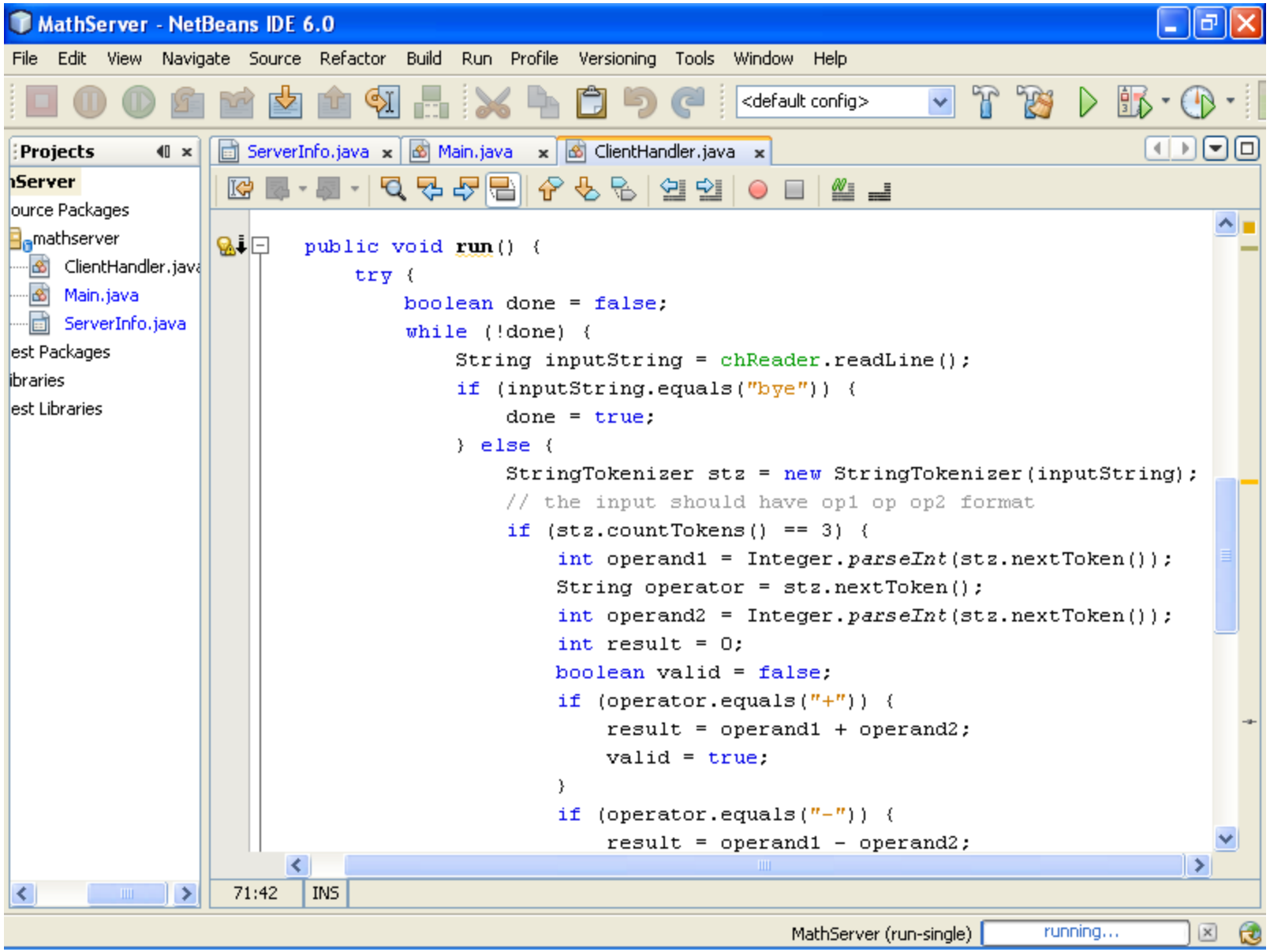

Host Name: 192.168.1.100
Port Number: 2115
Close

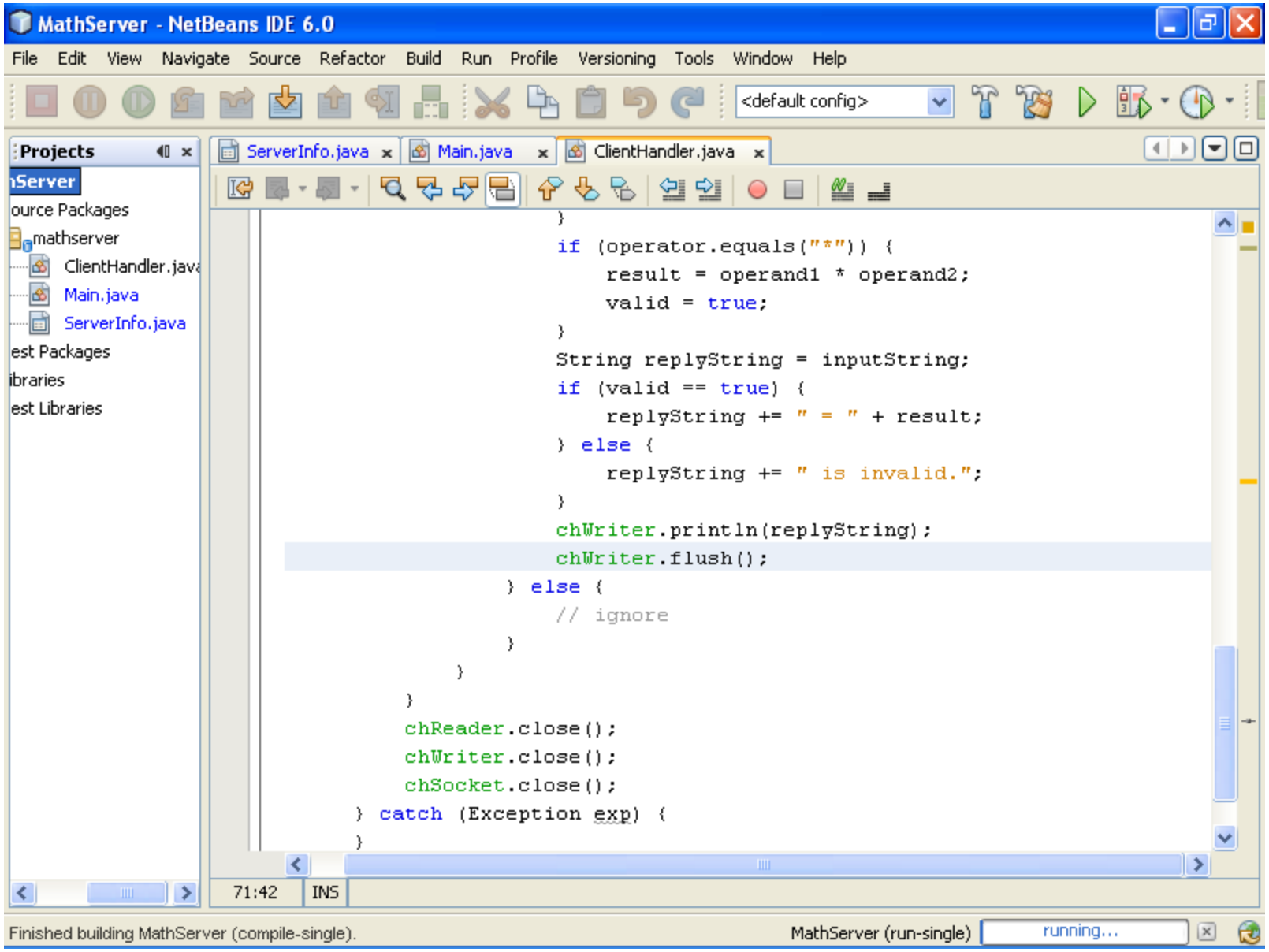

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
MathServer
Source Packages
Test Packa
Libraries
Test Librari
ServerInfo.java
Main.java
New
Find... Ctrl+F
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Delete
Refactor
Reverse Engineer...
Compile Package F9
Local History
Tools
JFrame Form...
JApplet...
HTML File...
JPanel Form...
Java Class...
Java Package...
Java Interface...
Entity Class...
Entity Classes from Database...
Web Service Client...
Other...
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
ServerInfo si = new ServerInfo();
si.setVisible(true);
}
}
11:4
INS

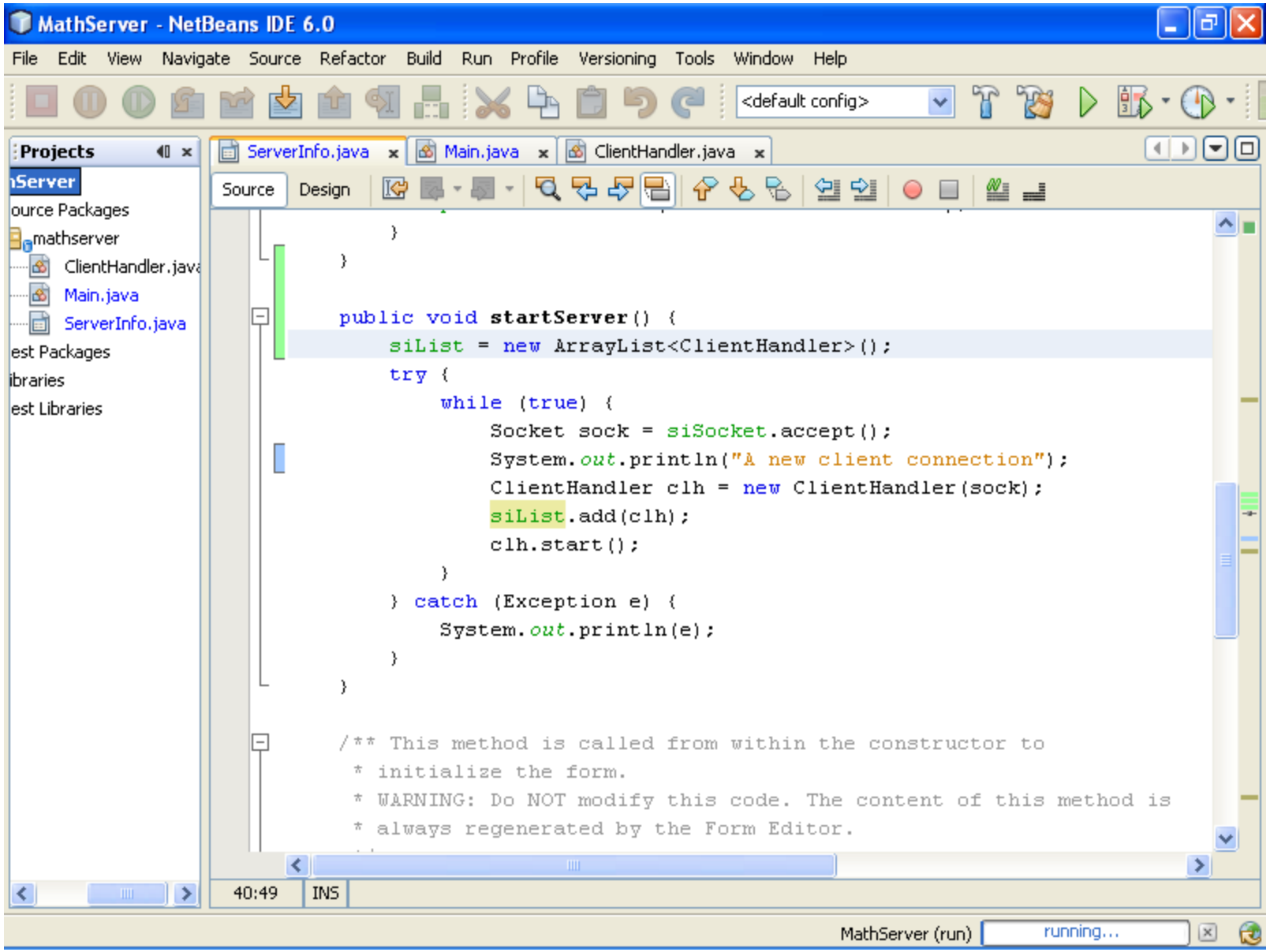

New Java Class
Steps
1. Choose File Type
2. Name and Location
Name and Location
Class Name: ClientHandler
Project: MathServer
Location: Source Packages
Package: mathserver
Created File: C:\yunglu\java\MathServer\src\mathserver\ClientHandler.jav
< Back
Next >
Finish
Cancel
Help

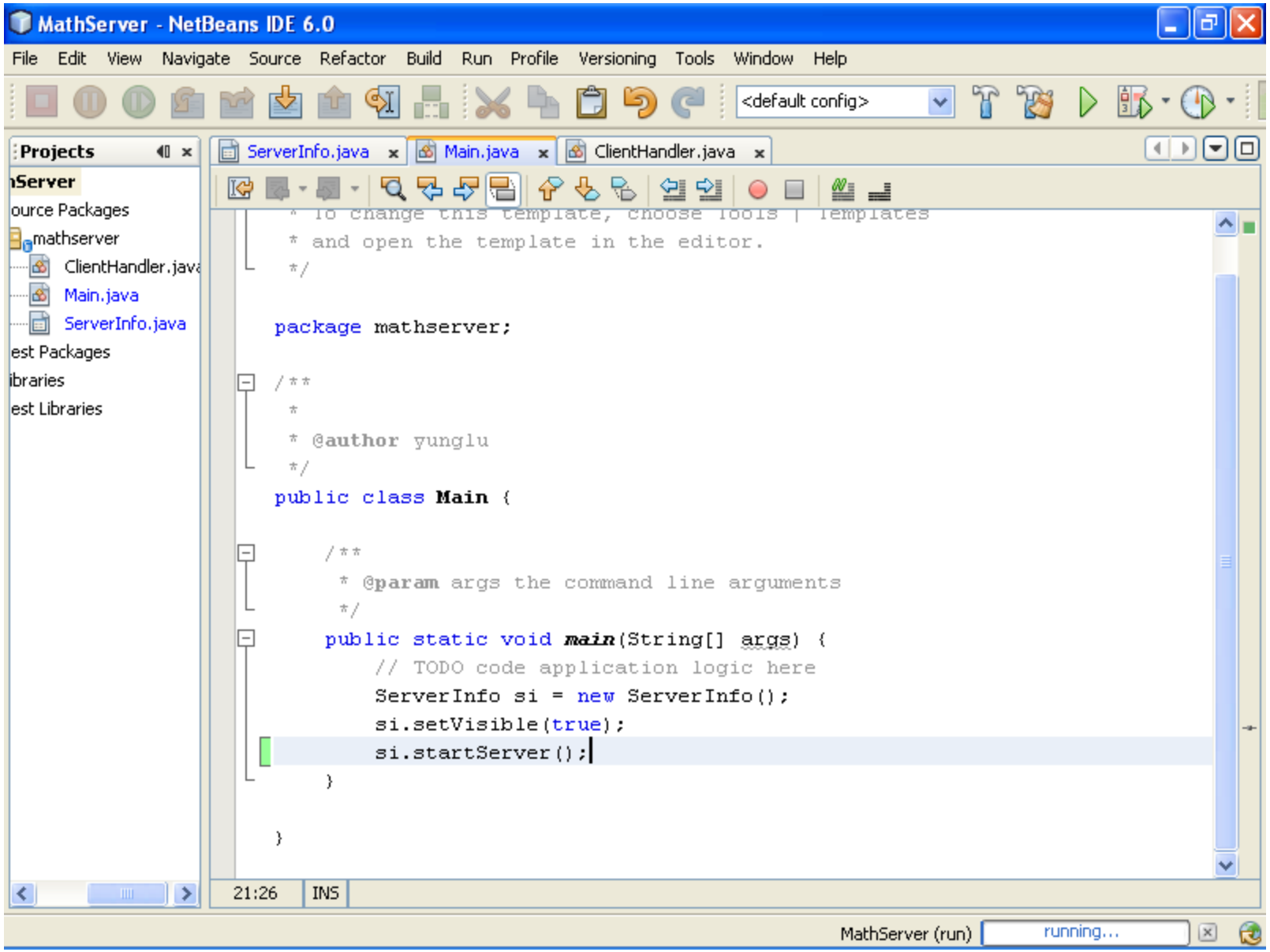
Handle Client Requests


MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
Server
ource Packages
mathserver
ClientHandler.java
Main.java
ServerInfo.java
est Packages
ibraries
est Libraries
ServerInfo.java
Main.java
ClientHandler.java
olic class ClientHandler extends Thread {
Socket chSocket;
private PrintWriter chWriter;
private BufferedReader chReader;
public ClientHandler(Socket sock) {
try {
chSocket = sock;
chWriter = new PrintWriter(sock.getOutputStream());
InputStream in_stream = sock.getInputStream();
InputStreamReader in_reader =
new InputStreamReader(in_stream);
chReader = new BufferedReader(in_reader);
} catch (Exception exp) {
}
}
public void run() {
try {
boolean done = false;
while (!done) {
71:42
INS
MathServer (run-single)
running...

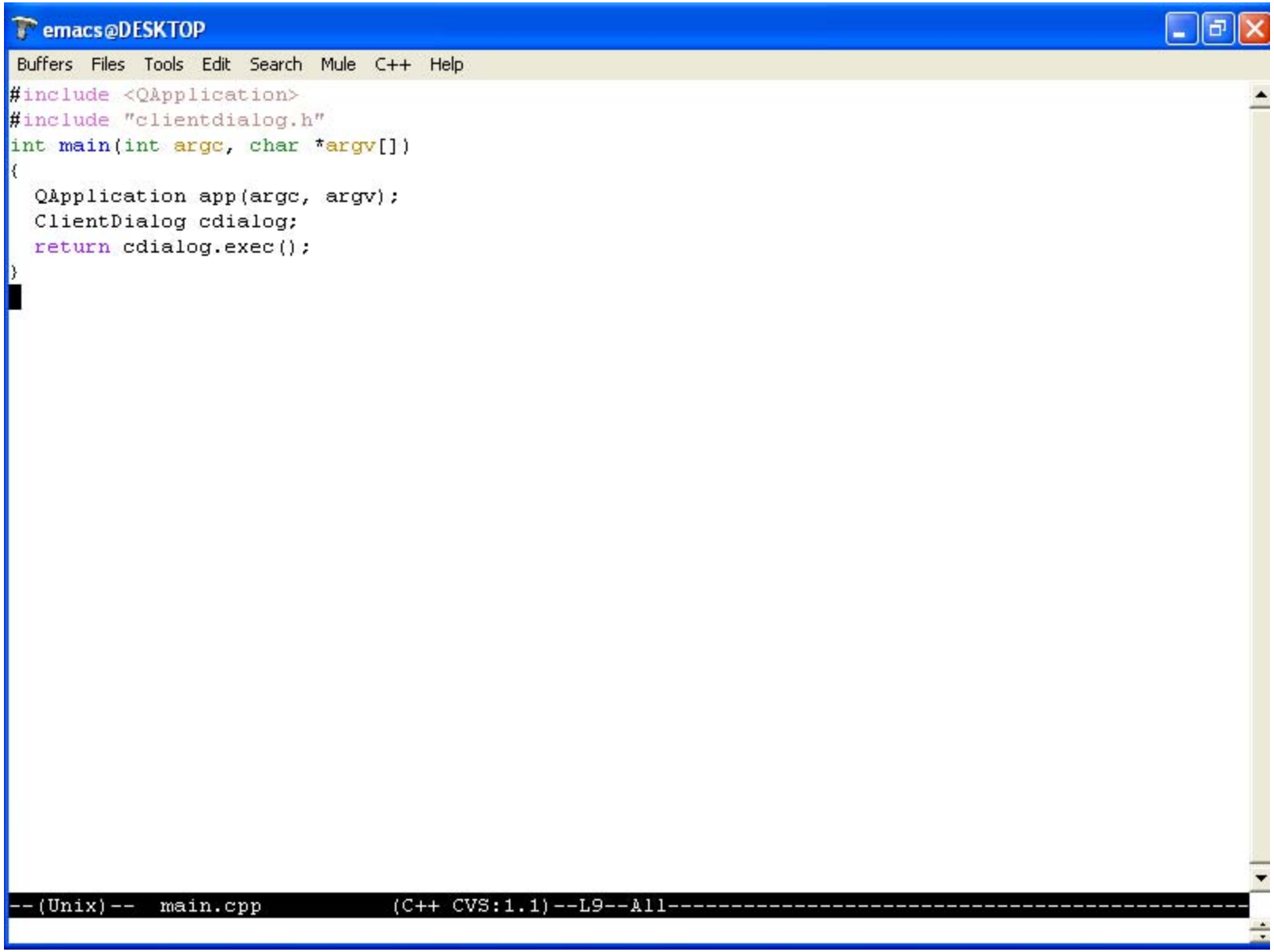

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
Server
ource Packages
mathserver
ClientHandler.java
Main.java
ServerInfo.java
est Packages
ibraries
est Libraries
ServerInfo.java
Main.java
ClientHandler.java
public void run() {
try {
boolean done = false;
while (!done) {
String inputString = chReader.readLine();
if (inputString.equals("bye")) {
done = true;
} else {
StringTokenizer stz = new StringTokenizer(inputString);
// the input should have op1 op op2 format
if (stz.countTokens() == 3) {
int operand1 = Integer.parseInt(stz.nextToken());
String operator = stz.nextToken();
int operand2 = Integer.parseInt(stz.nextToken());
int result = 0;
boolean valid = false;
if (operator.equals("+")) {
result = operand1 + operand2;
valid = true;
}
if (operator.equals("-")) {
result = operand1 - operand2;
71:42
INS
MathServer (run-single)
running...

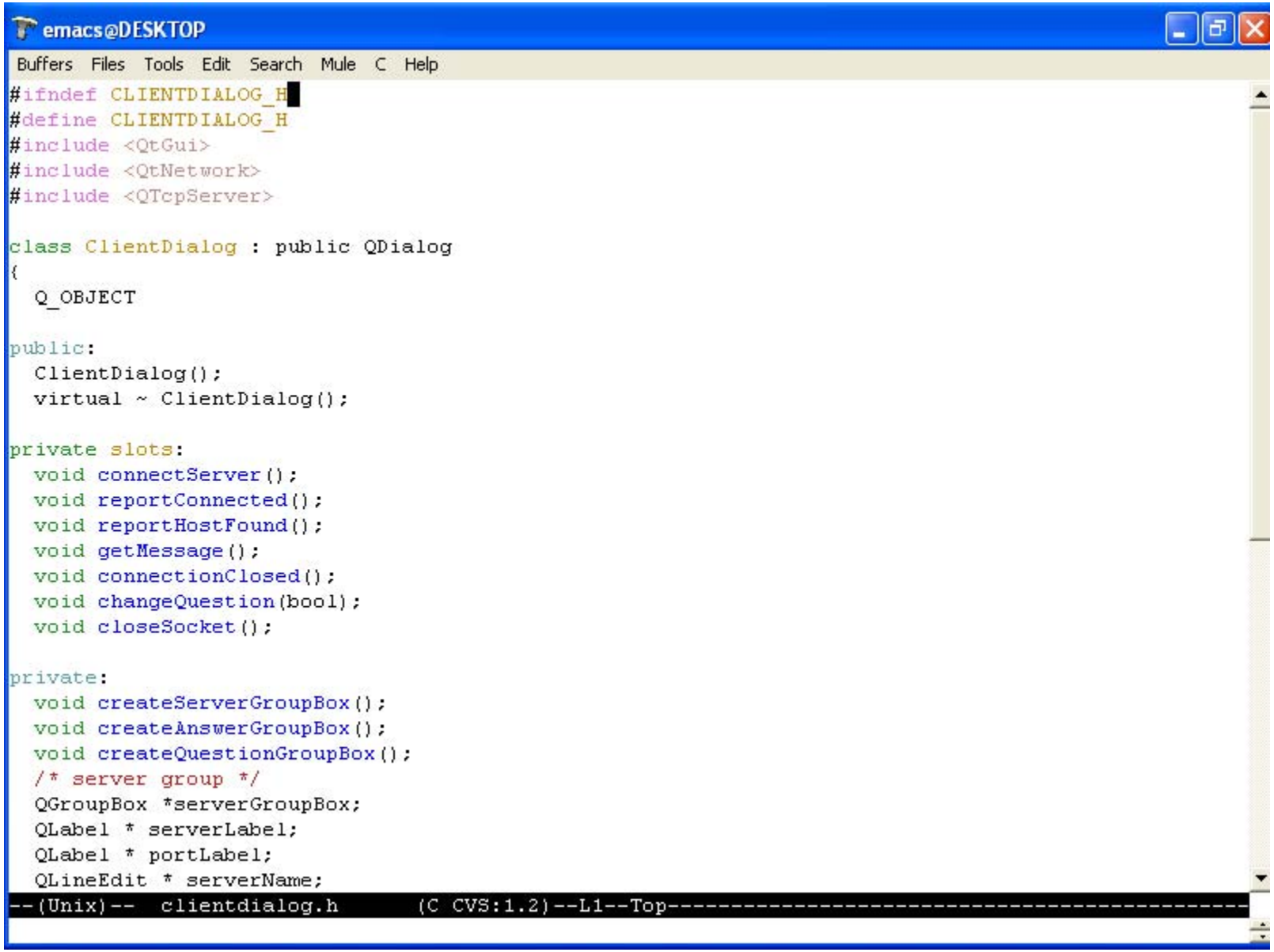

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
ServerInfo.java
Main.java
ClientHandler.java
ource Packages
mathserver
ClientHandler.java
Main.java
ServerInfo.java
est Packages
ibraries
est Libraries
} if (operator.equals("*")) { result = operand1 * operand2; valid = true; } String replyString = inputString; if (valid == true) { replyString += " = " + result; } else { replyString += " is invalid."; } chWriter.println(replyString); chWriter.flush(); } else { // ignore } } } chReader.close(); chWriter.close(); chSocket.close(); } catch (Exception exp) { }
71:42
INS
Finished building MathServer (compile-single).
MathServer (run-single)
running...

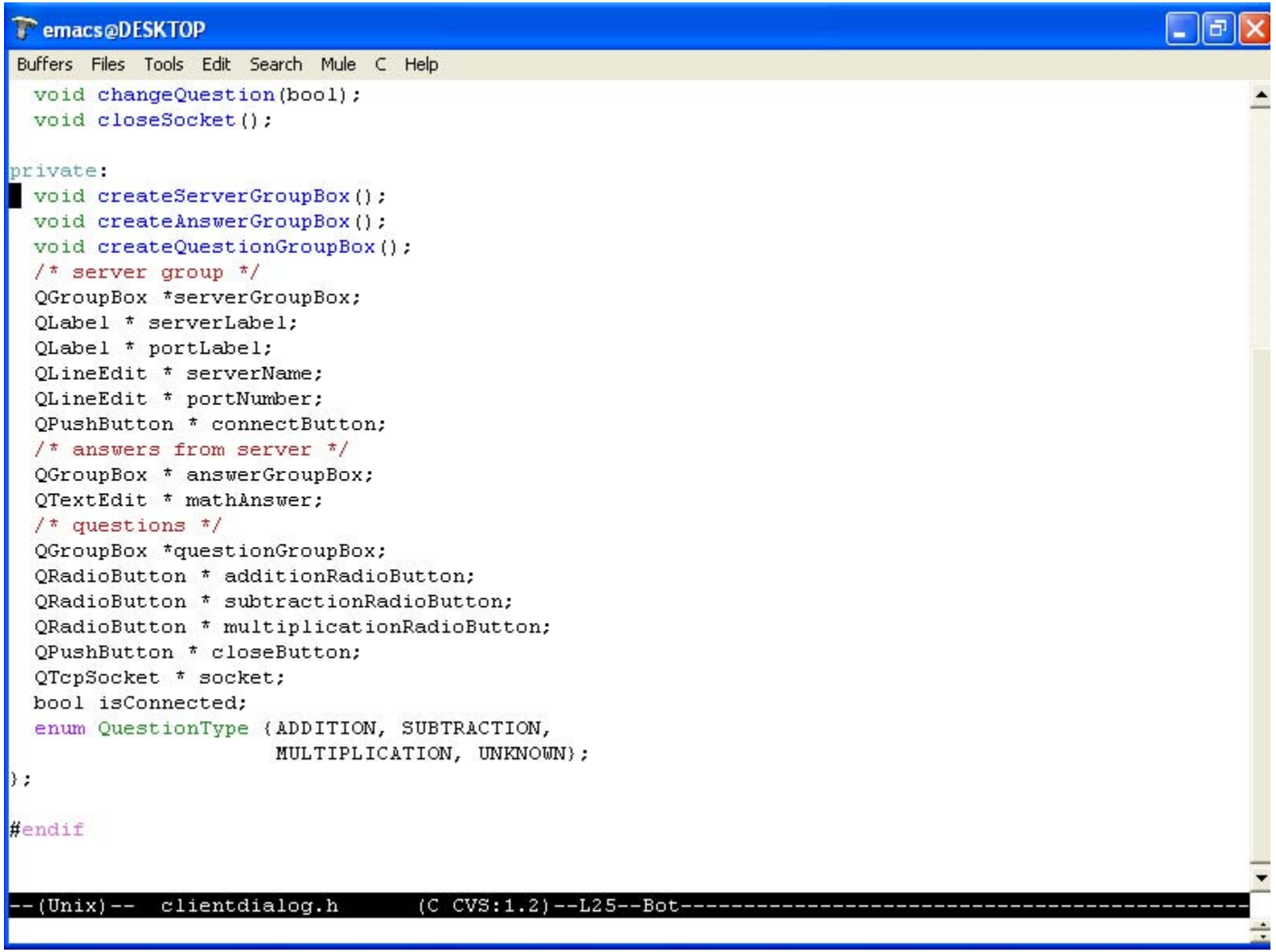

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
ServerInfo.java
Main.java
ClientHandler.java
Source Design
public void startServer() {
siList = new ArrayList<ClientHandler>();
try {
while (true) {
Socket sock = siSocket.accept();
System.out.println("A new client connection");
ClientHandler clh = new ClientHandler(sock);
siList.add(clh);
clh.start();
} catch (Exception e) {
System.out.println(e);
/** This method is called from within the constructor to
* initialize the form.
* WARNING: Do NOT modify this code. The content of this method is
* always regenerated by the Form Editor.
40:49
INS
MathServer (run)
running...

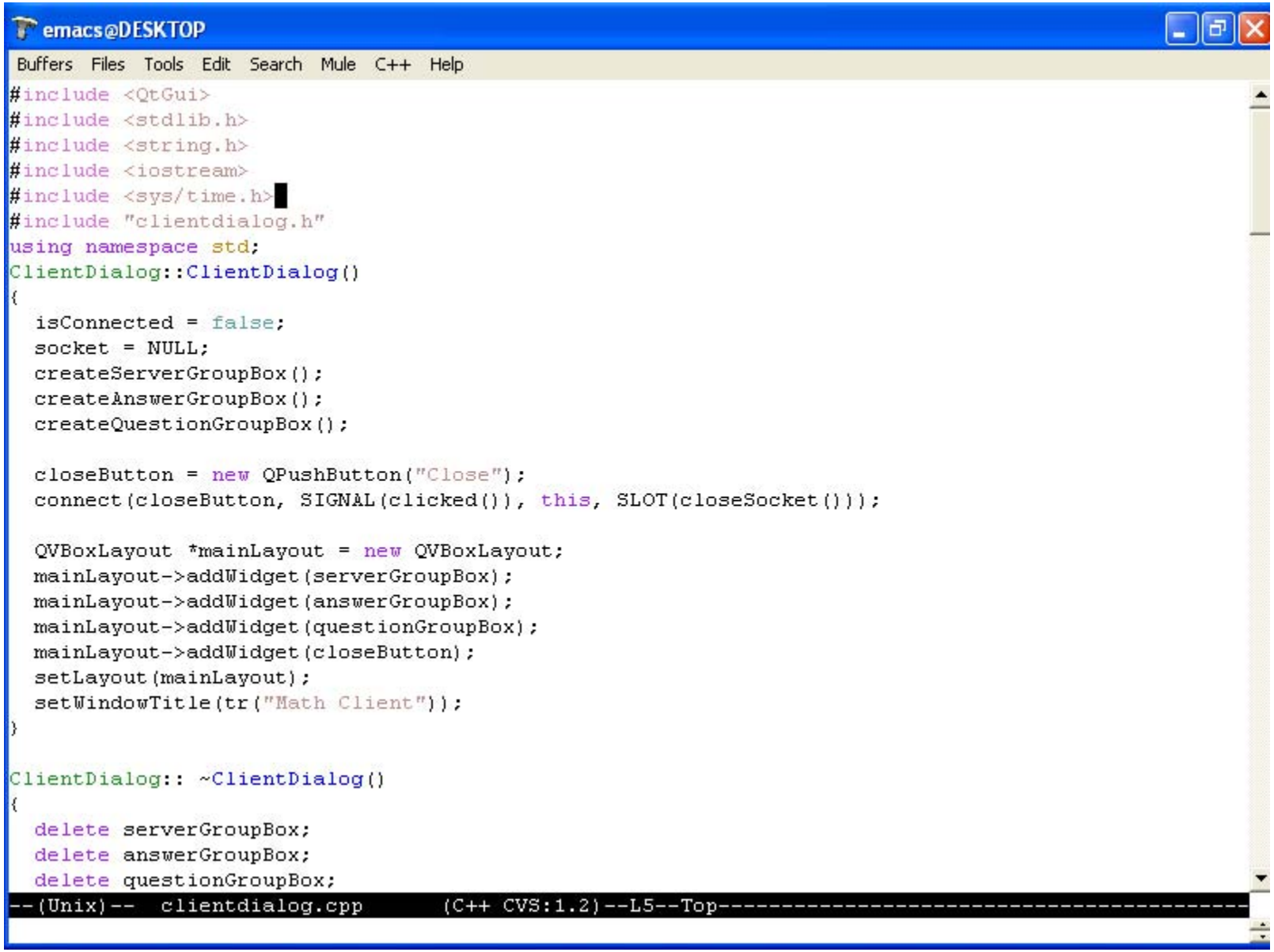

MathServer - NetBeans IDE 6.0
File Edit View Navigate Source Refactor Build Run Profile Versioning Tools Window Help
<default config>
Projects
ServerInfo.java
Main.java
ClientHandler.java
ource Packages
mathserver
ClientHandler.java
Main.java
ServerInfo.java
est Packages
ibraries
est Libraries
* and open the template in the editor.
*/
package mathserver;
/**
*
* @author yunglu
*/
public class Main {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here
ServerInfo si = new ServerInfo();
si.setVisible(true);
si.startServer();
}
}
21:26
INS
MathServer (run)
running...

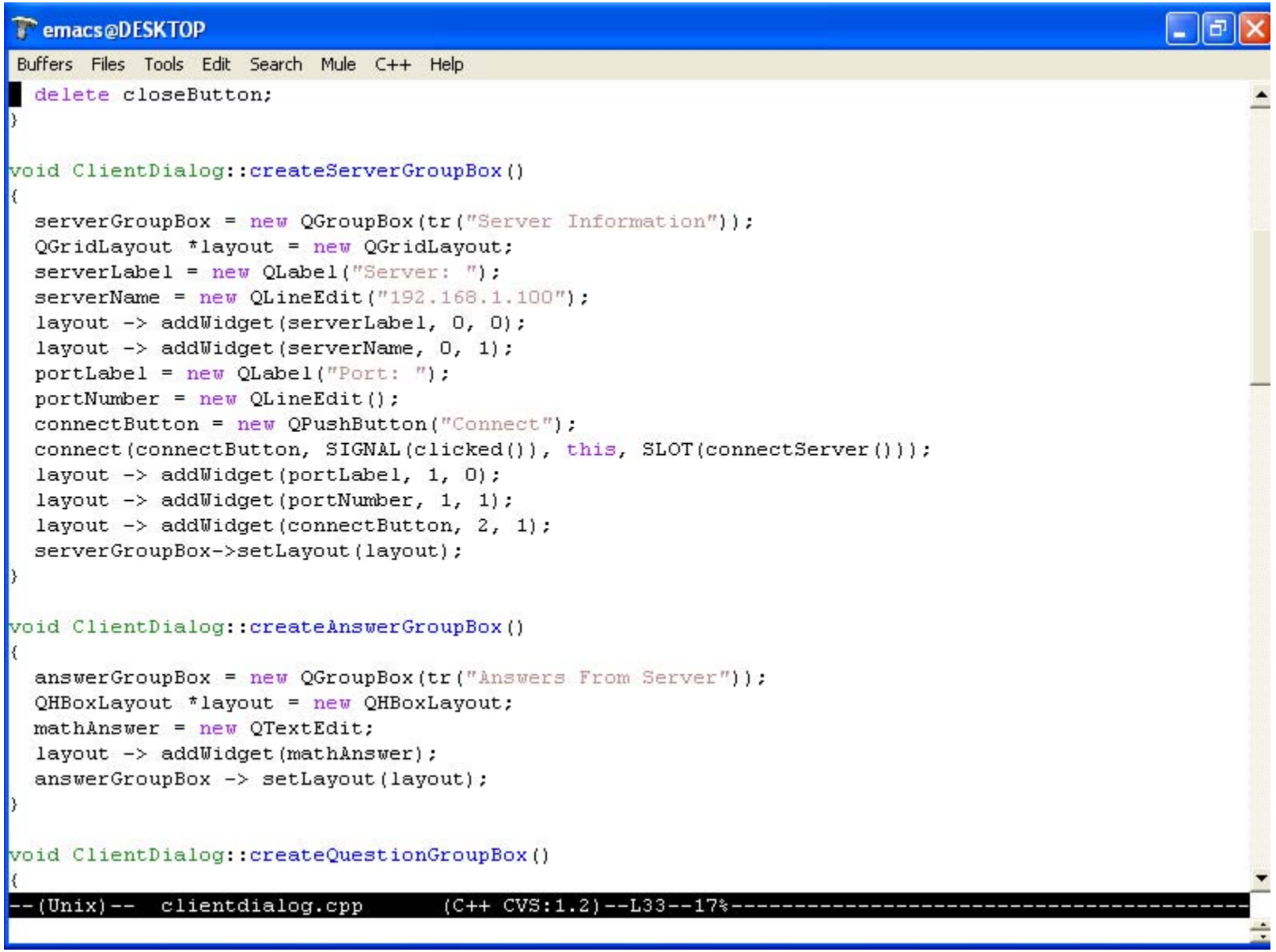
C++ Client

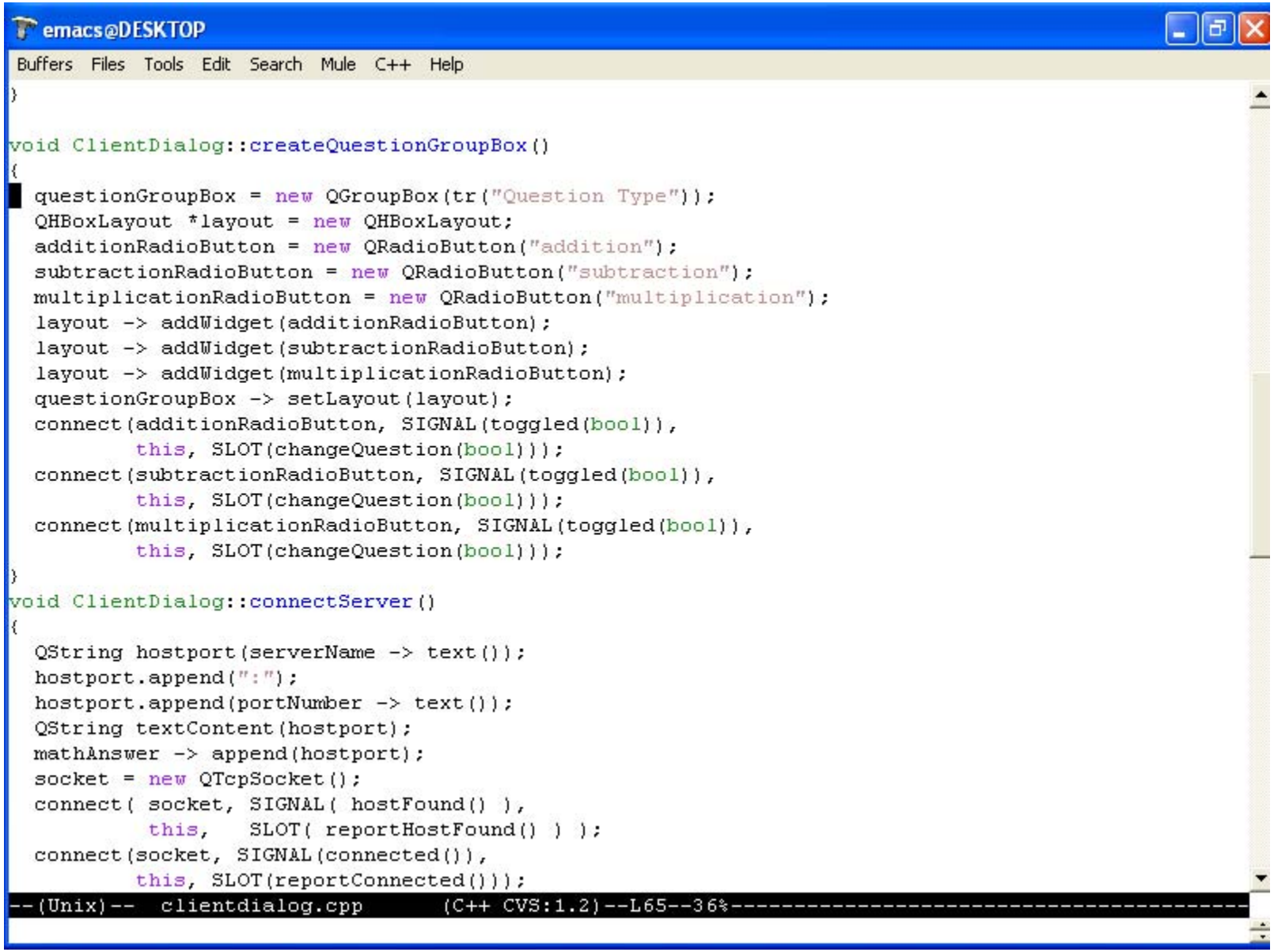

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
#include <QApplication>
#include "clientdialog.h"
int main(int argc, char *argv[])
{
QApplication app(argc, argv);
ClientDialog cdialog;
return cdialog.exec();
}
--(Unix)-- main.cpp (C++ CVS:1.1)--L9--All--

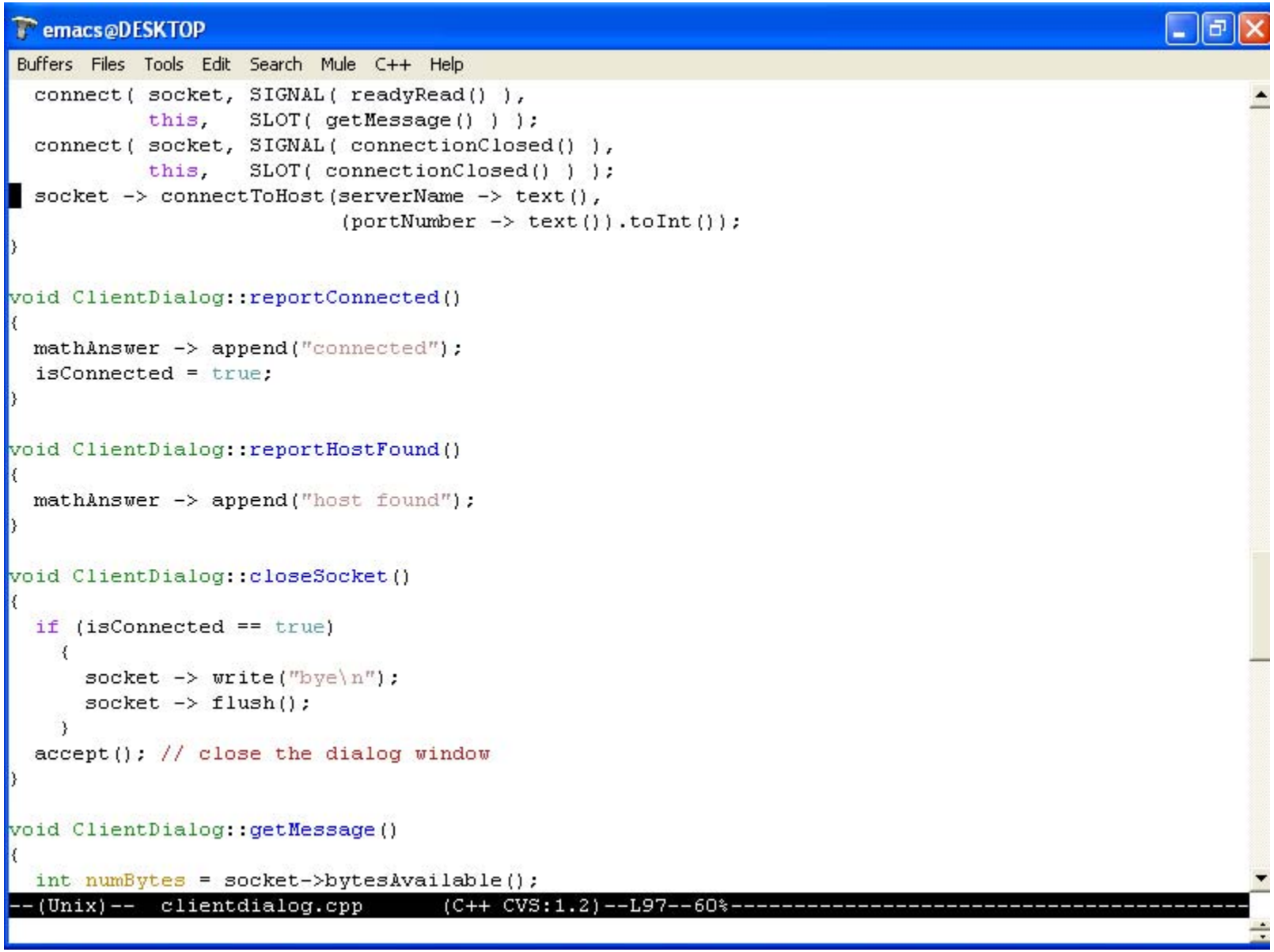

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C Help
#ifndef CLIENTDIALOG_H
#define CLIENTDIALOG_H
#include <QtGui>
#include <QtNetwork>
#include <QTcpServer>
class ClientDialog : public QDialog
{
Q_OBJECT
public:
ClientDialog();
virtual ~ ClientDialog();
private slots:
void connectServer();
void reportConnected();
void reportHostFound();
void getMessage();
void connectionClosed();
void changeQuestion(bool);
void closeSocket();
private:
void createServerGroupBox();
void createAnswerGroupBox();
void createQuestionGroupBox();
/* server group */
QGroupBox *serverGroupBox;
QLabel * serverLabel;
QLabel * portLabel;
QLineEdit * serverName;
--(Unix)-- clientdialog.h (C CVS:1.2)--L1--Top--

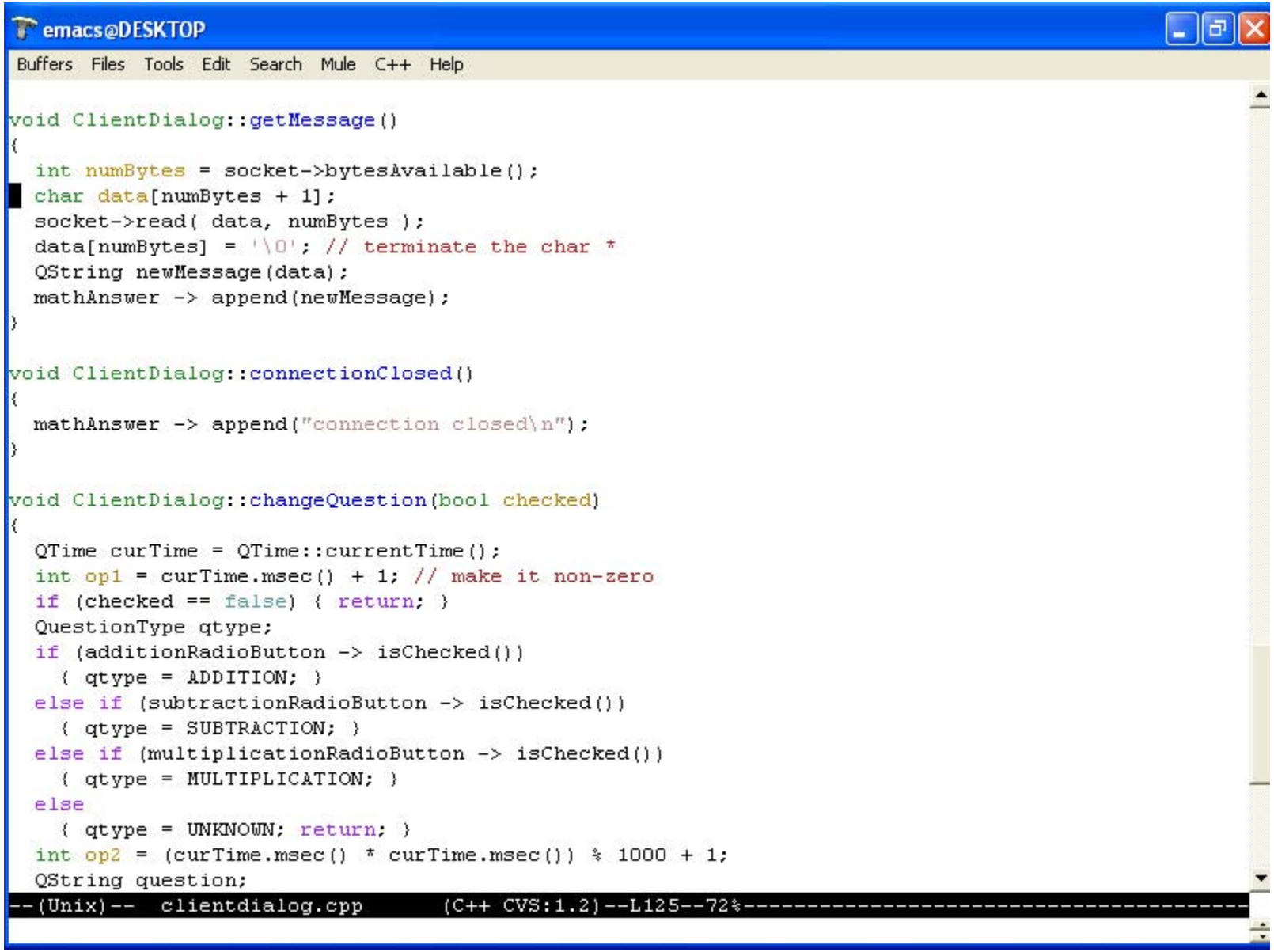

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C Help
void changeQuestion(bool);
void closeSocket();
private:
void createServerGroupBox();
void createAnswerGroupBox();
void createQuestionGroupBox();
/* server group */
QGroupBox *serverGroupBox;
QLabel * serverLabel;
QLabel * portLabel;
QLineEdit * serverName;
QLineEdit * portNumber;
QPushButton * connectButton;
/* answers from server */
QGroupBox * answerGroupBox;
QTextEdit * mathAnswer;
/* questions */
QGroupBox *questionGroupBox;
QRadioButton * additionRadioButton;
QRadioButton * subtractionRadioButton;
QRadioButton * multiplicationRadioButton;
QPushButton * closeButton;
QTcpSocket * socket;
bool isConnected;
enum QuestionType {ADDITION, SUBTRACTION,
MULTIPLICATION, UNKNOWN};
};
#endif
--(Unix)-- clientdialog.h (C CVS:1.2)--L25--Bot

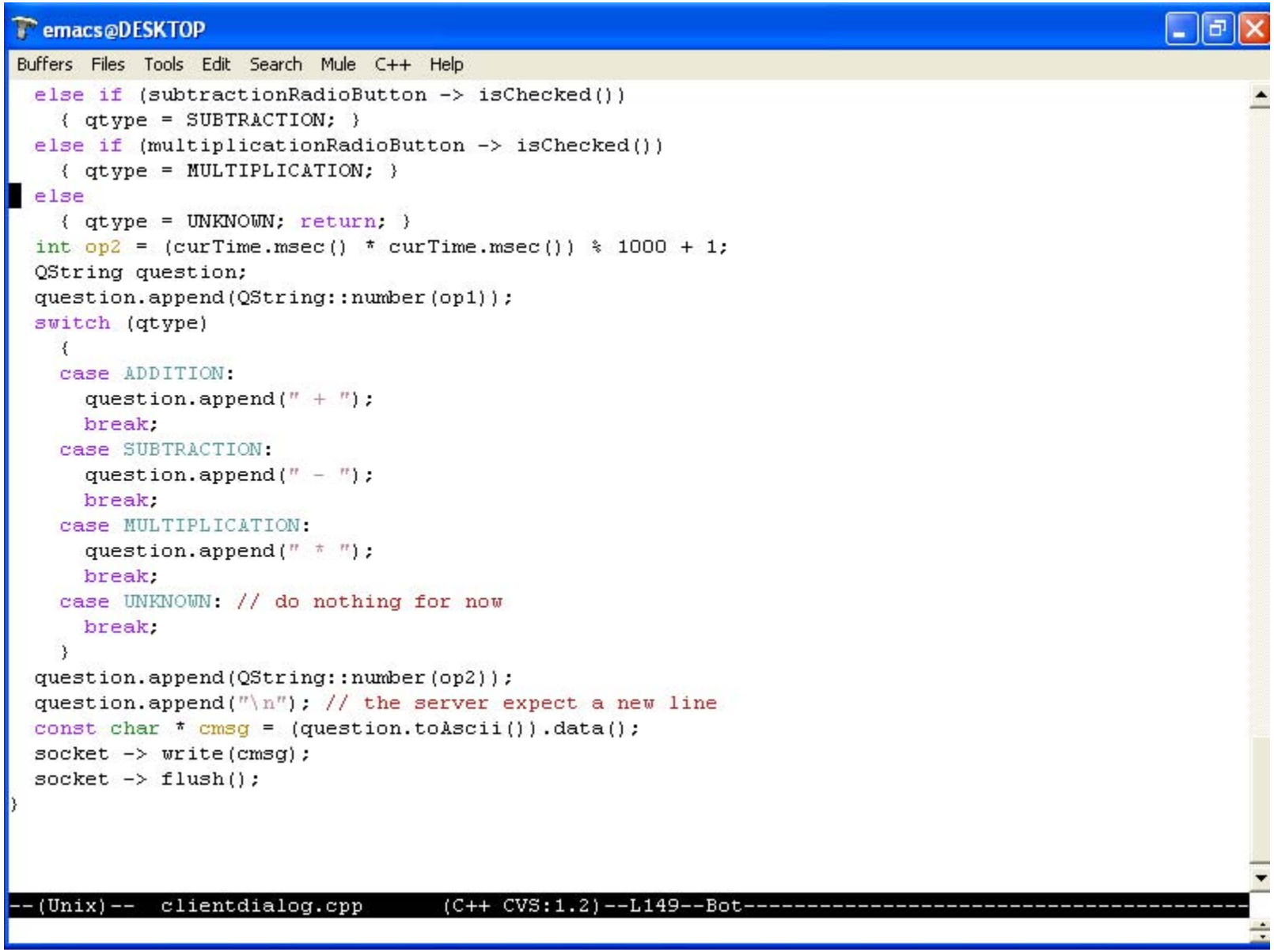

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
#include <QtGui>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include <sys/time.h>
#include "clientdialog.h"
using namespace std;
ClientDialog::ClientDialog()
{
isConnected = false;
socket = NULL;
createServerGroupBox();
createAnswerGroupBox();
createQuestionGroupBox();
closeButton = new QPushButton("Close");
connect(closeButton, SIGNAL(clicked()), this, SLOT(closeSocket()));
QVBoxLayout *mainLayout = new QVBoxLayout;
mainLayout->addWidget(serverGroupBox);
mainLayout->addWidget(answerGroupBox);
mainLayout->addWidget(questionGroupBox);
mainLayout->addWidget(closeButton);
setLayout(mainLayout);
setWindowTitle(tr("Math Client"));
}
ClientDialog:: ~ClientDialog()
{
delete serverGroupBox;
delete answerGroupBox;
delete questionGroupBox;
--(Unix)-- clientdialog.cpp (C++ CVS:1.2)--L5--Top--

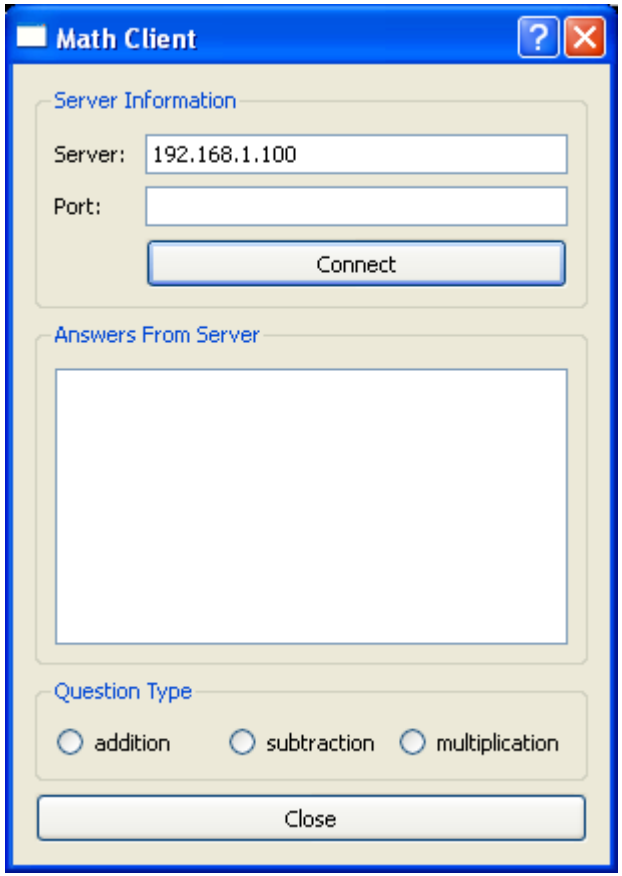

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
delete closeButton;
}
void ClientDialog::createServerGroupBox()
{
serverGroupBox = new QGroupBox(tr("Server Information"));
QGridLayout *layout = new QGridLayout;
serverLabel = new QLabel("Server: ");
serverName = new QLineEdit("192.168.1.100");
layout -> addWidget(serverLabel, 0, 0);
layout -> addWidget(serverName, 0, 1);
portLabel = new QLabel("Port: ");
portNumber = new QLineEdit();
connectButton = new QPushButton("Connect");
connect(connectButton, SIGNAL(clicked()), this, SLOT(connectServer()));
layout -> addWidget(portLabel, 1, 0);
layout -> addWidget(portNumber, 1, 1);
layout -> addWidget(connectButton, 2, 1);
serverGroupBox->setLayout(layout);
}
void ClientDialog::createAnswerGroupBox()
{
answerGroupBox = new QGroupBox(tr("Answers From Server"));
QHBoxLayout *layout = new QHBoxLayout;
mathAnswer = new QTextEdit;
layout -> addWidget(mathAnswer);
answerGroupBox -> setLayout(layout);
}
void ClientDialog::createQuestionGroupBox()
{
--(Unix)-- clientdialog.cpp (C++ CVS:1.2)--L33--17%

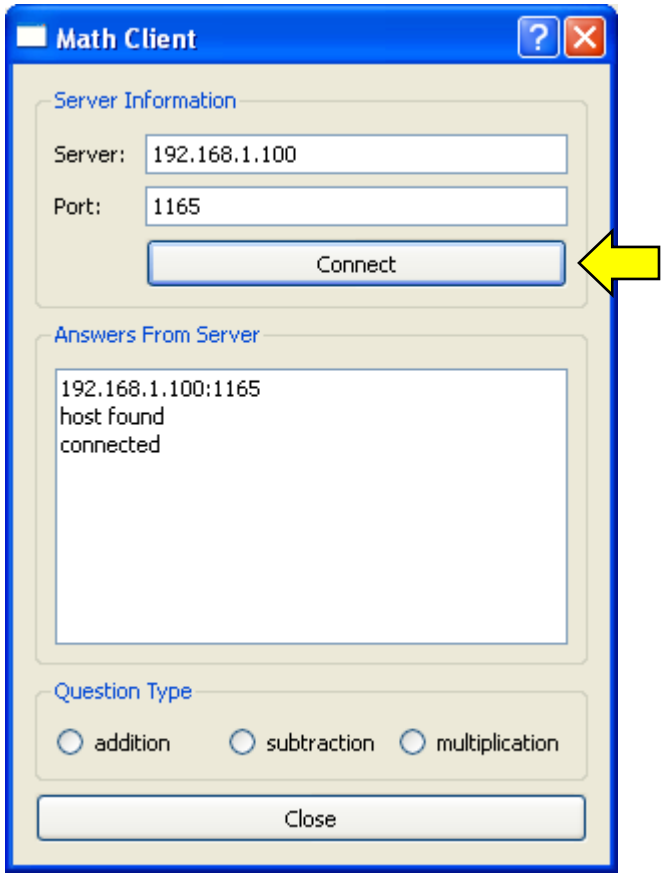

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
}
void ClientDialog::createQuestionGroupBox()
{
questionGroupBox = new QGroupBox(tr("Question Type"));
QHBoxLayout *layout = new QHBoxLayout;
additionRadioButton = new QRadioButton("addition");
subtractionRadioButton = new QRadioButton("subtraction");
multiplicationRadioButton = new QRadioButton("multiplication");
layout -> addWidget(additionRadioButton);
layout -> addWidget(subtractionRadioButton);
layout -> addWidget(multiplicationRadioButton);
questionGroupBox -> setLayout(layout);
connect(additionRadioButton, SIGNAL(toggled(bool)),
this, SLOT(changeQuestion(bool)));
connect(subtractionRadioButton, SIGNAL(toggled(bool)),
this, SLOT(changeQuestion(bool)));
connect(multiplicationRadioButton, SIGNAL(toggled(bool)),
this, SLOT(changeQuestion(bool)));
}
void ClientDialog::connectServer()
{
QString hostport(serverName -> text());
hostport.append(":");
hostport.append(portNumber -> text());
QString textContent(hostport);
mathAnswer -> append(hostport);
socket = new QTcpSocket();
connect(socket, SIGNAL(hostFound()),
this, SLOT(reportHostFound()));
connect(socket, SIGNAL(connected()),
this, SLOT(reportConnected()));
--(Unix)-- clientdialog.cpp (C++ CVS:1.2)--L65--36%

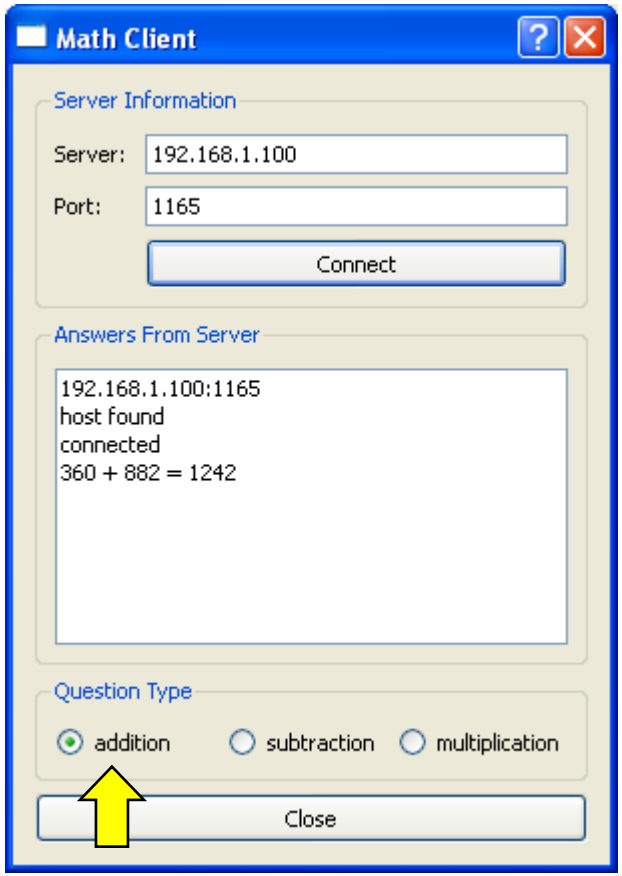

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
connect(socket, SIGNAL(readyRead()),
this, SLOT(getMessage()));
connect(socket, SIGNAL(connectionClosed()),
this, SLOT(connectionClosed()));
socket -> connectToHost(serverName -> text(),
(portNumber -> text()).toInt());
}
void ClientDialog::reportConnected()
{
mathAnswer -> append("connected");
isConnected = true;
}
void ClientDialog::reportHostFound()
{
mathAnswer -> append("host found");
}
void ClientDialog::closeSocket()
{
if (isConnected == true)
{
socket -> write("bye\n");
socket -> flush();
}
accept(); // close the dialog window
}
void ClientDialog::getMessage()
{
int numBytes = socket->bytesAvailable();
--(Unix)-- clientdialog.cpp (C++ CVS:1.2)--L97--60%

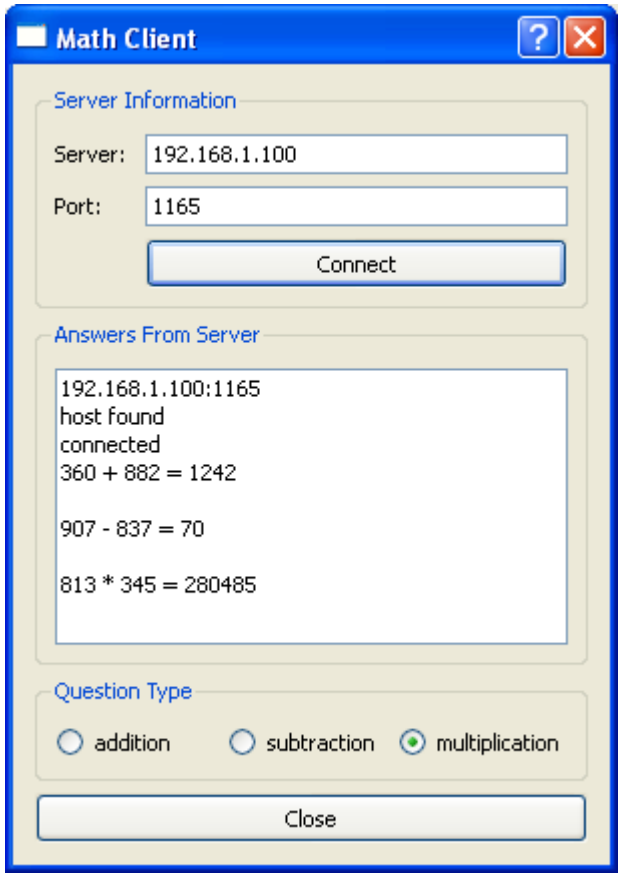

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
void ClientDialog::getMessage()
{
int numBytes = socket->bytesAvailable();
char data[numBytes + 1];
socket->read(data, numBytes);
data[numBytes] = '\0'; // terminate the char *
QString newMessage(data);
mathAnswer -> append(newMessage);
}
void ClientDialog::connectionClosed()
{
mathAnswer -> append("connection closed\n");
}
void ClientDialog::changeQuestion(bool checked)
{
QTime curTime = QTime::currentTime();
int op1 = curTime.msec() + 1; // make it non-zero
if (checked == false) { return; }
QuestionType qtype;
if (additionRadioButton -> isChecked())
{ qtype = ADDITION; }
else if (subtractionRadioButton -> isChecked())
{ qtype = SUBTRACTION; }
else if (multiplicationRadioButton -> isChecked())
{ qtype = MULTIPLICATION; }
else
{ qtype = UNKNOWN; return; }
int op2 = (curTime.msec() * curTime.msec()) % 1000 + 1;
QString question;
--(Unix)-- clientdialog.cpp (C++ CVS:1.2)--L125--72%

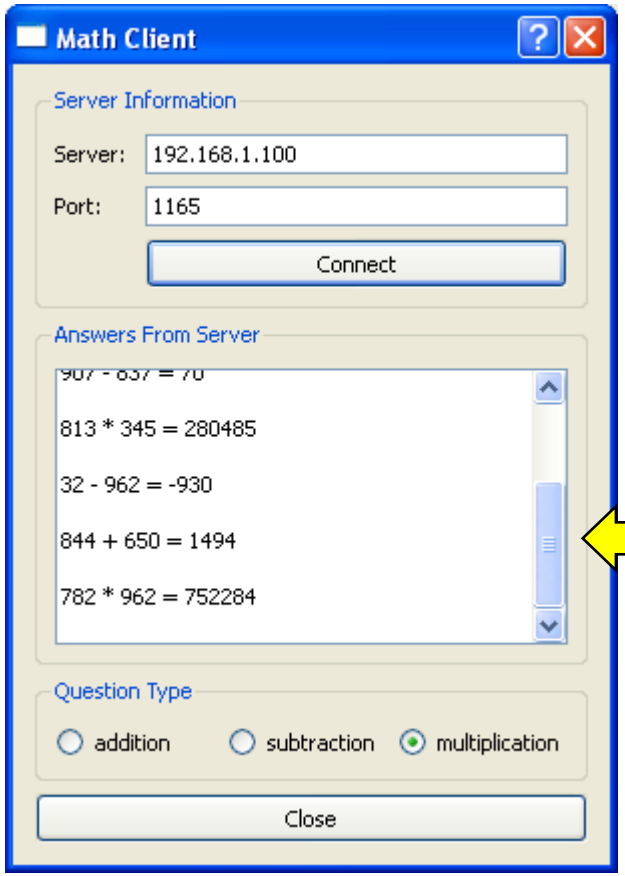

emacs@DESKTOP
Buffers Files Tools Edit Search Mule C++ Help
else if (subtractionRadioButton -> isChecked())
{ qtype = SUBTRACTION; }
else if (multiplicationRadioButton -> isChecked())
{ qtype = MULTIPLICATION; }
else
{ qtype = UNKNOWN; return; }
int op2 = (curTime.msec() * curTime.msec()) % 1000 + 1;
QString question;
question.append(QString::number(op1));
switch (qtype)
{
case ADDITION:
question.append(" + ");
break;
case SUBTRACTION:
question.append(" - ");
break;
case MULTIPLICATION:
question.append(" * ");
break;
case UNKNOWN: // do nothing for now
break;
}
question.append(QString::number(op2));
question.append("\n"); // the server expect a new line
const char * cmsg = (question.toAscii()).data();
socket -> write(cmsg);
socket -> flush();
}
--(Unix)-- clientdialog.cpp (C++ CVS:1.2)--L149--Bot


Math Client
Server Information
Server:
192.168.1.100
Port:
Connect
Answers From Server
Question Type
addition
subtraction
multiplication
Close

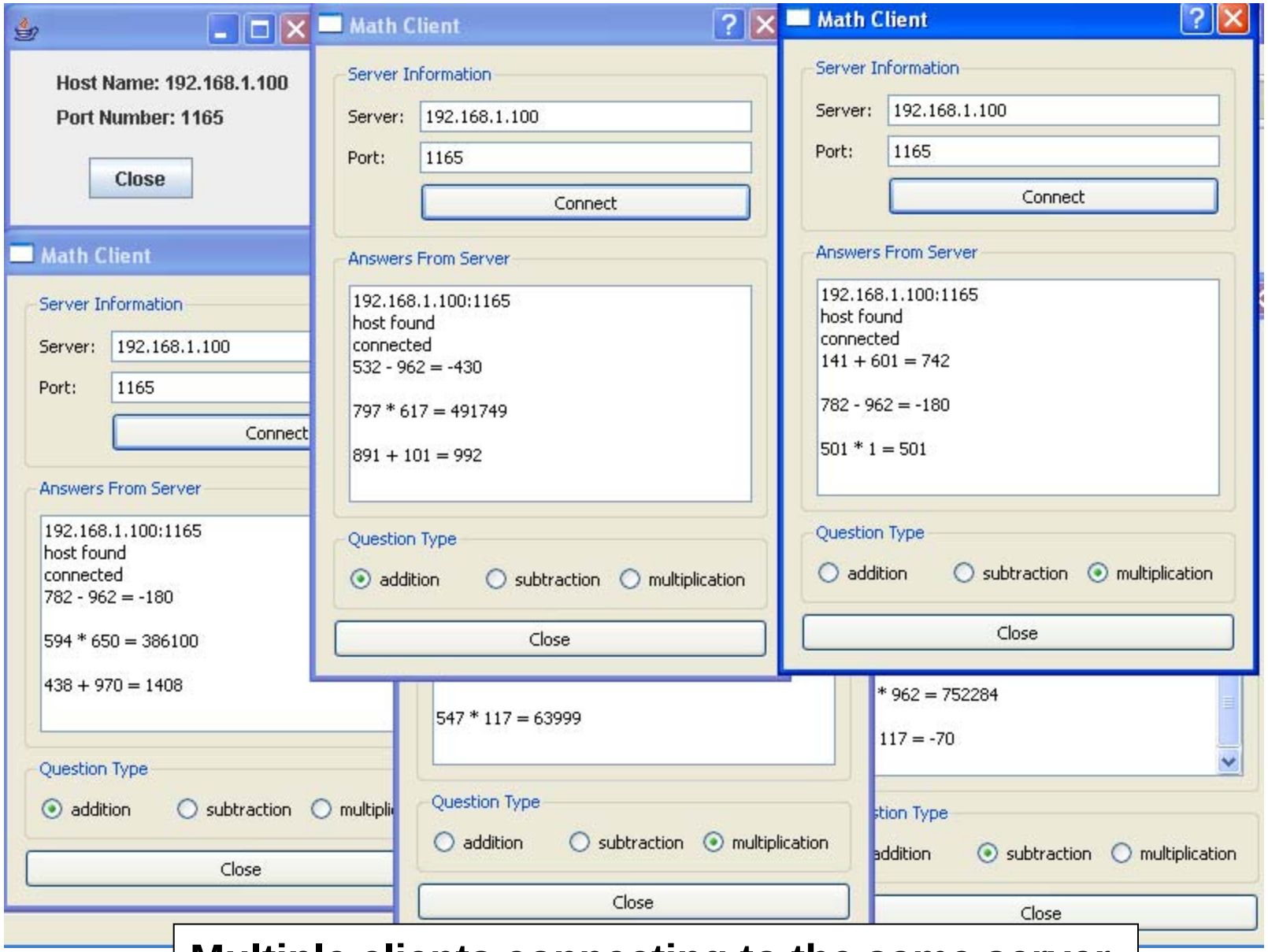

Math Client
Server Information
Server:
192.168.1.100
Port:
1165
Connect
Answers From Server
192.168.1.100:1165
host found
connected
Question Type
addition
subtraction
multiplication
Close


Math Client
Server Information
Server:
192.168.1.100
Port:
1165
Connect
Answers From Server
192.168.1.100:1165
host found
connected
360 + 882 = 1242
Question Type
addition
subtraction
multiplication
Close


Math Client
Server Information
Server:
192.168.1.100
Port:
1165
Connect
Answers From Server
192.168.1.100:1165
host found
connected
360 + 882 = 1242
907 - 837 = 70
813 * 345 = 280485
Question Type
addition
subtraction
multiplication
Close



Scrollbar automatically added



Multiple clients connecting to the same server.

The C++ program **must not leak memory**.
valgrind –tool=memcheck to detect memory leak

Submission: A zip file of the CVS repository

Remember to commit all changes first.

Submit this exercise only.

Do not submit any other exercise.

Do not submit a wrong zip file.