

Re: Class.getMethod in class's static initializer block

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-08/msg00106.html>

- *From:* Mark Space <markspace@xxxxxxxxxxxxxxxx>
 - *Date:* Wed, 01 Aug 2007 21:44:39 GMT
-

chucky wrote:

```
Method m = map.get(str);
if(m != null)
m.invoke();
else{ /* default code */ }
```

You seem to have got the info you needed already. I just wanted to add that I had some similar code for a "command line server" project. You might find some good ideas in the code below, or I might get criticized for language abuse. Either way it'll be educational for someone. ^_^

```
/*
 * ServerTest.java
 *
 * Created on July 13, 2007, 4:28 PM
 *
 * To change this template, choose Tools | Template Manager
 * and open the template in the editor.
 */
```

```
package servertest;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.HashMap;
import java.util.Map;
import java.util.logging.ConsoleHandler;
import java.util.logging.Level;
```

Re: Class.getMethod in class's static initializer block

```
import java.util.logging.Logger;

/**
 *
 * @author B
 */

/**
 * Copyright 2007 All rights reserved.
 */

public class ServerTest implements Runnable
{
    static private final boolean DEBUG = true;

    public static void main(String[] args)
    {
        // TODO code application logic here
        new Thread( new ServerTest() ).start();
    }

    // DEFAULT CONSTRUCTOR
    /** Creates a new instance of ServerTest */
    public ServerTest()
    {
        if( logger == null )
        {
            logger = Logger.getLogger( ServerTest.class.getName() );
            if( DEBUG )
            {
                ConsoleHandler ch = new ConsoleHandler();
                ch.setLevel( Level.ALL );
                logger.addHandler( ch );
                logger.setLevel( Level.ALL );
                logger.setUseParentHandlers( false );
            }
        }
        // port = 25; // 25 is telnet port #
        port = 4040;
        runServer = true;
        commandList = new HashMap<String, CommandLine>();
        addCommand( new QuitLineCommand() );
        addCommand( new HelloLineCommand() );
        addCommand( new ErrorLineCommand() );
        addCommand( new TestLineCommand( "test", new CommandRunner()
        { public ReturnCode runCommand( String s )
        { return (ReturnCode) new TestReturnCode( 0, false, "Test succeeded." ); } } ));
    }

    public void addCommand( CommandLine cl )
```

Re: Class.getMethod in class's static initializer block

```
{
commandList.put( cl.getCommand(), cl );
}

public void run()
{
while( runServer )
{
try
{
ServerSocket sock = new ServerSocket( port );
logger.info( "Accepting connections on port " + port );
// System.err.println( "Accepting connections on port " + port );

Socket client = sock.accept();

InputStream clientInput = client.getInputStream();
OutputStream clientOutput = client.getOutputStream();

PrintWriter clientPW = new PrintWriter( clientOutput, true );
BufferedReader clientBR = new BufferedReader( new InputStreamReader( clientInput ) );

while( true )
{
String line = clientBR.readLine();
// System.err.println( line );
logger.finer( line );
if( line.length() == 0 )
{
continue;
}
String commands [] = line.split( ":", 2 );
// System.err.print( commands.length + " commands " );
logger.log( Level.FINEST, "Split commands are ", commands );
// for( int i = 0; i < commands.length; i++ )
// {
// System.err.print( commands[i] + ", " );
// }
// System.err.println();

CommandLine exeObj = null;
if( commands.length >= 1 )
{
exeObj = commandList.get( commands[0] );
}
if( exeObj != null )
{
// System.err.println( exeObj );
String arguments = null;
if( commands.length > 1 )
{
```

Re: Class.getMethod in class's static initializer block

```
arguments = commands[1];
}
ReturnCode rc = exeObj.runCommand( commands[0], arguments );
if( rc == null )
{
// System.err.println( "return object was null!" );
logger.warning( "Return object was null!" );
continue;
}
if( rc.isError() )
{
clientPW.print( '\07' ); // ASCII bell
}
if( rc.getCode() != 0 )
{
clientPW.print( "(" + rc.getCode() + ")" );

}
if( rc.getMessage() != null )
{
clientPW.print( rc.getMessage() );
}
clientPW.println();
if( rc.getCode() == -1 )
{
runServer = false;
break;
}
else
{
clientPW.println( "\07Command not found." ); // \07 is ASCII bell
// clientPW.flush();
}
}
catch (IOException ex)
{
ex.printStackTrace();
}
}
}

private int port;
private boolean runServer;
private Map<String, CommandLine> commandList;
private static Logger logger;
```

Re: Class.getMethod in class's static initializer block

```
}

class LineParser
{
public boolean addCommand( CommandLine c )
{
//...
return true;
}

public ReturnCode parseLine( String line )
{
ReturnCode rc = null;
//...
return rc;
}
}

interface CommandLine
{
// public CommandLine( String command );
public String getCommand();
public ReturnCode runCommand( String command, String args );
}

interface CommandRunner
{
public ReturnCode runCommand( String args );
}

class TestLineCommand implements CommandLine
{
public TestLineCommand( String command, CommandRunner exec )
{
this.command = command;
this.exec = exec;
}

public String getCommand()
{
return command;
}

public ReturnCode runCommand(String command, String args)
{
return exec.runCommand( args );
}

private String command;
private CommandRunner exec;
}


```

Re: Class.getMethod in class's static initializer block

```
class QuitLineCommand implements CommandLine
{
public String getCommand()
{
return "quit";
}

public ReturnCode runCommand(String command, String args)
{
return (ReturnCode) new TestReturnCode( -1, false, "Good-bye" );
}
}

class ErrorLineCommand implements CommandLine
{
public String getCommand()
{
return "error";
}

public ReturnCode runCommand(String command, String args)
{
return (ReturnCode) new TestReturnCode( 1, true, args );
}
}

class HelloLineCommand implements CommandLine
{
public String getCommand()
{
return "hello";
}

public ReturnCode runCommand( String command, String args )
{
return (ReturnCode) new TestReturnCode( 0, false, "Hello World! You typed: " + args );
}
}

interface ReturnCode
{
public int getCode();
public boolean isError();
public String getMessage();
// public String getExtendedMessage();
}

class TestReturnCode implements ReturnCode
{
public TestReturnCode( int code, boolean isError, String message )
```

Re: Class.getMethod in class's static initializer block

```
{
this.code = code;
this.error = isError;
this.message = message;
}

public int getCode()
{
return code;
}

public boolean isError()
{
return error;
}

public String getMessage()
{
return message;
}

private int code;
private boolean error;
private String message;
}
```

.