import java.util;

/**
* A ShellCommandTable object maintains a dispatch table of
* ShellCommand objects keyed by the command names used to invoke
* them.
*
* To add a new shell command to the table, install it from
* method fillTable().
*
* @see ShellCommand
*
* @version 7
*/
public class ShellCommandTable
{
private Map table = new TreeMap();

/**
* Construct and fill a shell command table.
*/
public ShellCommandTable()
{
fillTable();
}

/**
* Get a ShellCommand, given the command name key.
*
* @param key the name associated with the command we're
* looking for.
*
* @return the command we're looking for, null if none.
*/
public ShellCommand lookup( String key )
{
ShellCommand commandObject = (ShellCommand) table.get( key );
if (commandObject != null) {
return commandObject;
}

// try to load dynamically
// construct classname = "KeyCommand"
char[] chars = (key + "Command").toCharArray();
chars[0] = key.toUpperCase().charAt(0);
String classname = new String(chars);
try {
commandObject = (ShellCommand)Class.forName(classname).newInstance();
}
catch (Exception e) { // couldn't find class
return null;
}
install(key, commandObject); // put it in table for next time
return commandObject;
}

/**
* Get an array of the command names.
*
* @return the array of command names.
*/
public String[] getCommandNames()
{
return (String[]) table.keySet().toArray( new String[0] );
}

private void install( String commandName, ShellCommand command )
{
table.put( commandName, command );
}

private void fillTable()
{
install( "list", new ListCommand() );
install( "cd", new CdCommand() );
install( "newfile", new NewfileCommand() );
install( "remove", new RemoveCommand() );
install( "help", new HelpCommand() );
install( "mkdir", new MkdirCommand() );
install( "type", new TypeCommand() );
install( "logout", new LogoutCommand() );
}