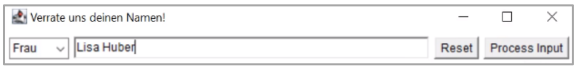
Beispiel 1:

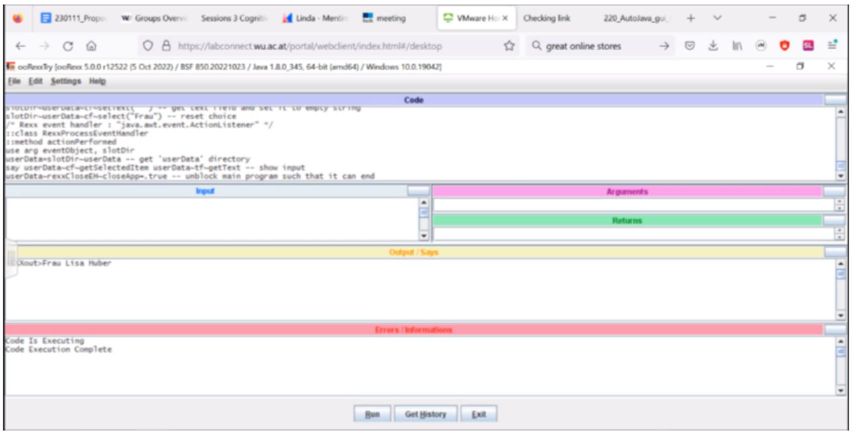
```
rexxCloseEH = .RexxCloseAppEventHandler~new -- Rexx event handler
rpCloseEH = BsfCreateRexxProxy(rexxCloseEH, , "java.awt.event.WindowListener")
f=.bsf~new("java.awt.Frame", "Verrate uns deinen Namen!") -- create frame
f~addWindowListener(rpCloseEH) -- add RexxProxy event handler
f~setLayout( .bsf~new("java.awt.FlowLayout") ) -- create FlowLayout object and assign it
userData = .directory~new -- a directory which will be passed to Rexx with the event
userData~rexxCloseEH=rexxCloseEH -- save Rexx event handler for later use
cf=.BSF~new("java.awt.Choice") -- create Choice object
userData~cf=cf -- add choice field for later use
cf ~~add("Frau") ~~add("Herr") ~~add("Divers")-- add options/choices
f~add(cf) -- add Choice object to frame
tf=.bsf~new("java.awt.TextField", "", 50) -- create TextField, show 50 chars
userData~tf=tf -- add text field for later use
f~add(tf) -- add TextField object to frame
but=.bsf~new('java.awt.Button', 'Reset') -- create Button object
f~add(but) -- add Button object to frame
rp=BsfCreateRexxProxy(.RexxResetEventHandler~new, userData, "java.awt.event.ActionListener")
but~addActionListener(rp) -- add RexxProxy event handler
but=.bsf~new('java.awt.Button', 'Process Input') -- create Button object
f~add(but) -- add Button object to frame
rp=BsfCreateRexxProxy(.RexxProcessEventHandler~new, userData, "java.awt.event.ActionListener")
but~addActionListener(rp) -- add RexxProxy event handler
f ~~pack ~~setVisible(.true)~~toFront -- layout the Frame object, show it, make sure it is in front
rexxCloseEH~waitForExit -- wait until we are allowed to end the program
call BSF.terminateRexxEngine -- inhibit callbacks from Java (necessary, if Rexx started Java)
::requires BSF.cls -- load Object Rexx BSF support
-- ... continued on next page ...
/* Rexx event handler to set "close app" indicator: "java.awt.event.WindowListener" */
::class RexxCloseAppEventHandler
::method init /* constructor */
expose closeApp -- used as control variable
closeApp = .false
::method windowClosing -- event method (from WindowListener)
expose closeApp
closeApp=.true -- change control variable to unblock
::method unknown -- intercept unhandled events, do nothing
::attribute closeApp -- allow to get and set the control variable's value
::method waitForExit -- blocking (waiting) method
expose closeApp
guard on when closeApp=.true-- blocks (waits) until control variable is set to .true
/* Rexx event handler: "java.awt.event.ActionListener" */
::class RexxResetEventHandler
::method actionPerformed
use arg eventObject, slotDir
slotDir~userData~tf~setText(" ") -- get text field and set it to empty string
slotDir~userData~cf~select("Frau") -- reset choice
/* Rexx event handler : "java.awt.event.ActionListener" */
::class RexxProcessEventHandler
::method actionPerformed
```

```
use arg eventObject, slotDir
userData=slotDir~userData -- get 'userData' directory
say userData~cf~getSelectedItem userData~tf~getText -- show input
userData~rexxCloseEH~closeApp=.true -- unblock main program such that it can end.
```



Frau

Lisa Huber

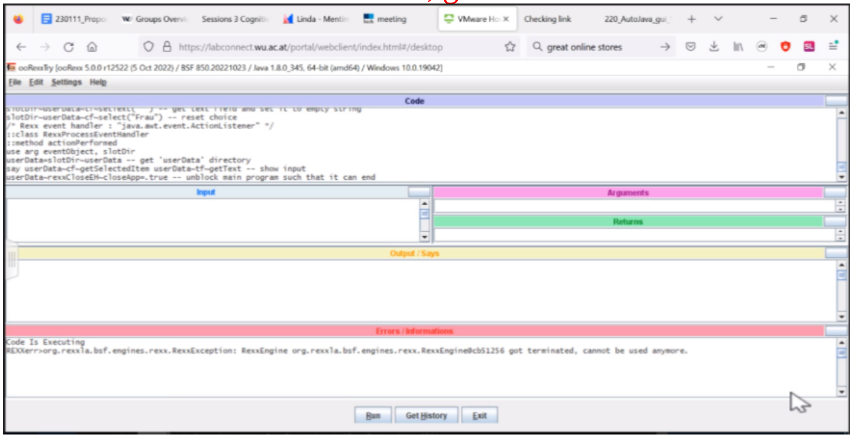

ooRexxTry [ooRexx 5.0.0 r12522 (5 Oct 2022) / BSF 850.20221023 / Java 1.8.0_345, 64-bit (amd64) / Windows 10.0.19042]
File
Edit
Settings
Help
Code
Input
Arguments
Returns
Output / Says
Errors / Informations
Run
Get History
Exit

```
slotDir~userData~tf~setText("") -- get text field and set it to empty string
slotDir~userData~cf~select("Frau") -- reset choice
/* Rexx event handler : "java.awt.event.ActionListener" */
::class RexxProcessEventHandler
::method actionPerformed
use arg eventObject, slotDir
userData=slotDir~userData -- get 'userData' directory
say userData~cf~getSelectedItem userData~tf~getText -- show input
userData~rexxCloseEH~closeApp=.true -- unblock main program such that it can end
```

```
Xout>Frau Lisa Huber
```

```
Code Is Executing
Code Execution Complete
```

Wenn wir es wiederholen wollen, geht es nicht mehr:

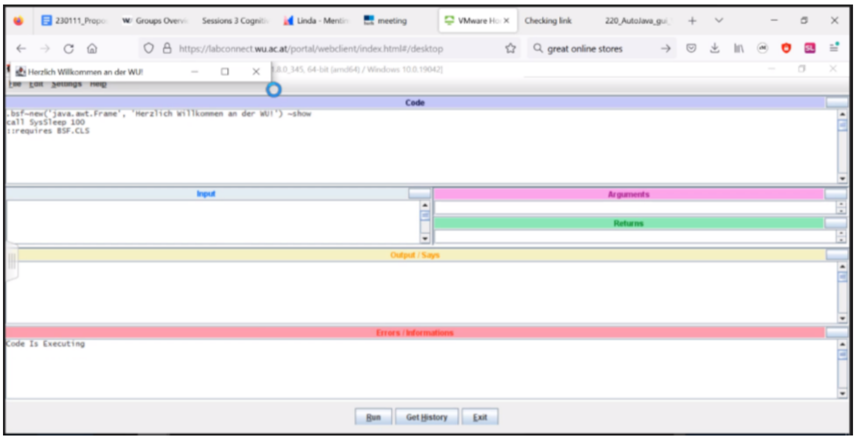

ooRexxTry [ooRexx 5.0.0 r12522 (5 Oct 2022) / BSF 850.20221023 / Java 1.8.0_345, 64-bit (amd64) / Windows 10.0.19042]
File
Edit
Settings
Help
Code
Input
Arguments
Returns
Output / Says
Errors / Informations
Run
Get History
Exit

```
slotDir~userData~tf~setText("") -- get text field and set it to empty string
slotDir~userData~cf~select("Frau") -- reset choice
/* Rexx event handler : "java.awt.event.ActionListener" */
::class RexxProcessEventHandler
::method actionPerformed
use arg eventObject, slotDir
userData=slotDir~userData -- get 'userData' directory
say userData~cf~getSelectedItem userData~tf~getText -- show input
userData~rexxCloseEH~closeApp=.true -- unblock main program such that it can end
```

```
Code Is Executing
REXXerr>org.rexxla.bsf.engines.rexx.RexxException: RexxEngine org.rexxla.bsf.engines.rexx.RexxEngine@cb51256 got terminated, cannot be used anymore.
```

Beispiel 2:

```
.bsf~new('java.awt.Frame', 'Herzlich Willkommen an der WU!') ~show
call SysSleep 100
::requires BSF.CLS
```



```
.bsf~new('java.awt.Frame', 'Herzlich Willkommen an der WU!') ~show
call SysSleep 100
::requires BSF.CLS
```

```
Code Is Executing
```