

G

G ehk graafiline programmeerimiskeel arenduskeskkonnas LabView

Indrek Sünter
2010

Möh?

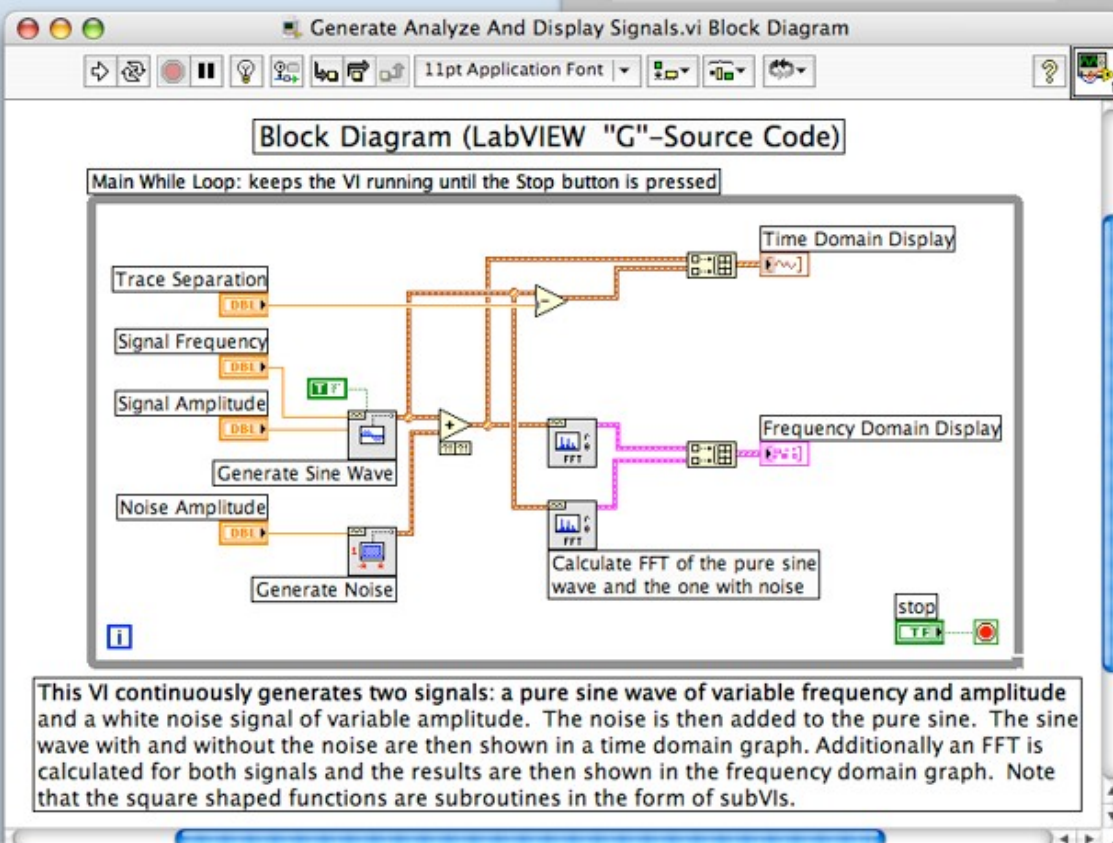
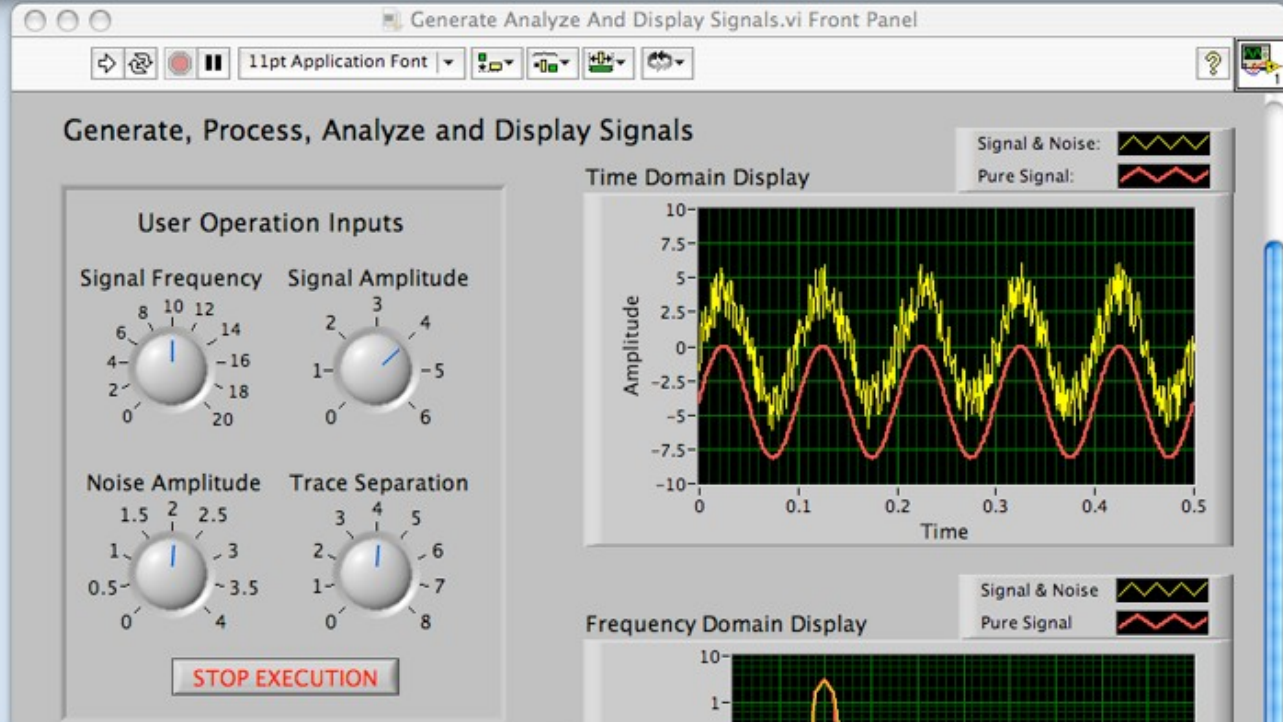
- **G** - visuaalne programmeerimiskeel firmalt National Instruments
- **LabView** - Laboratory Virtual Instrumentation Engineering Workbench
- Milleks?
 - Andmehõive arvutiga ühendatud mõõteseadmetelt
 - Arvutiga ühendatud seadmete juhtimine
 - Tööstusseadmete programmeerimiseks

Misasi see G on?

- 4G Andmevoo programmeerimiskeel
 - Programm kui suunatud graaf või plokk-diagramm
 - Plokk käivitatakse, kui kõik sisendid on väärtustatud
 - *Parallel processing* üksteisest eraldatud graafi osadel
- Loodud spetsiaalselt LabView keskkonnale, puuduvad alternatiivsed IDE-d
- Sõltumatus platvormist
 - Windows, Mac OS X, Linux
 - FPGA, ARM (PocketPC, PDA, mobiiltelefonid jpm.)

LabView *features*

- Tohutult draivereid igat sorti raudvara jaoks
- Madala- ja kõrgtasemeline ligipääs rauale
 - Real-Time
- Väga palju erinevaid tasuta ja tasulisi lisapakette VI-dega
- Keskkond LabView raudvaralise toega C/C++ keeles programmeerijatele (LabWindows/CVI)
- Vaba koodibaas näidetega ja aktiivne abistav kogukond
- Kompileerimine G koodist *stand-alone* programmiks

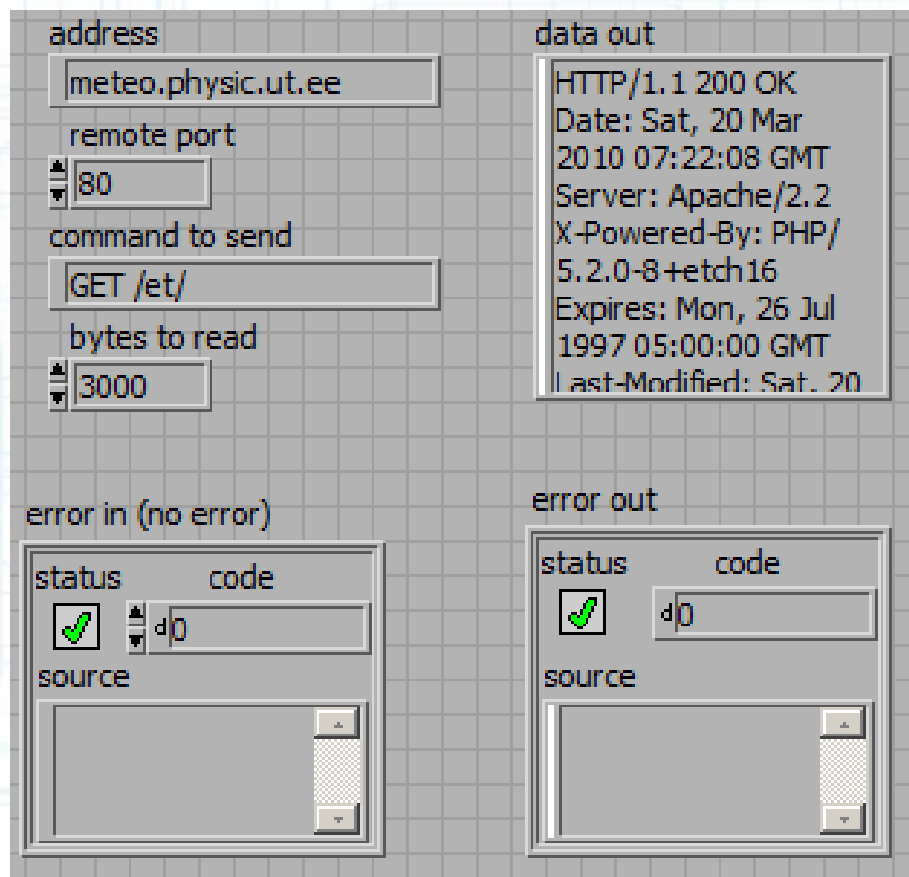


Timeline

- LabView 1.0 (Macintosh, 1986)
- LabWindows/CVI (DOS, 1987)
- LabView 2.0 (1990)
- LabView (Sun, Windows, 1992)
- LabView (Multiplatform, 1993)
- LabView Real-Time (1999)
- LabView 7 Express (2003)
- LabView 8.2 (OOP, 2006)
- LabView 9 (rekursioon, 32 ja 64-bit, 2009)

LabView - esipaneel

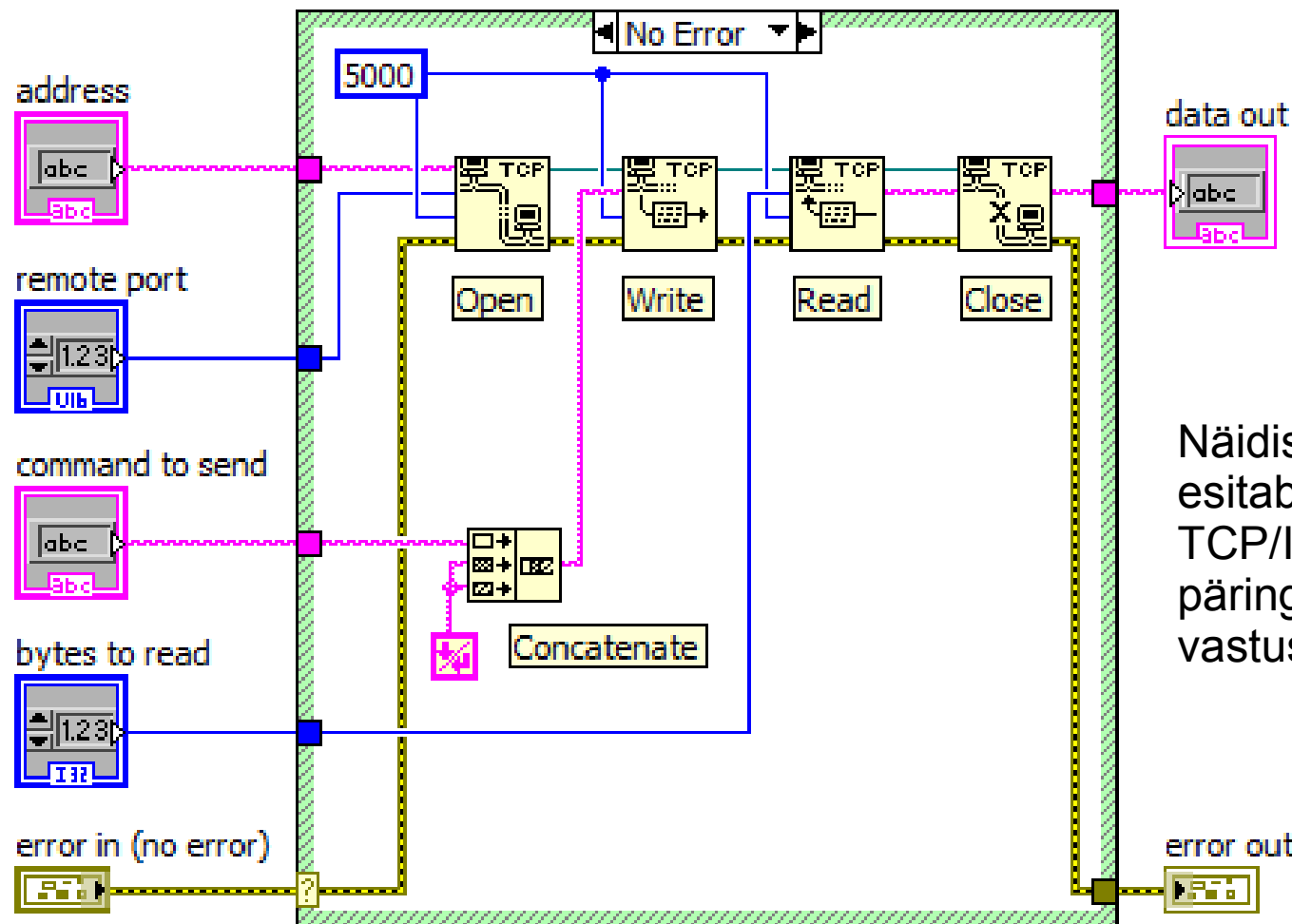
- GUI, mis kirjeldab sisendid (kontrollid) ja väljundid (indikaatorid)
- Nimed, selgitused ja vaikeväärtused sisenditel



Näidis-VI, mis esitab serverile TCP/IP protokollis päringu ning loeb vastuse.

LabView - plokk-diagramm

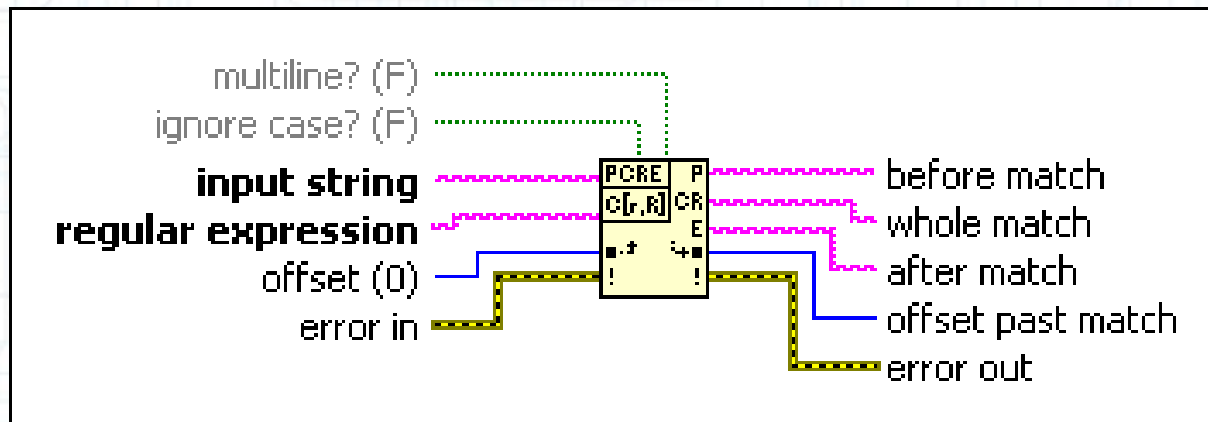
- Sisendeid ja väljundeid ühendav operatsioonide ahel



Näidis-VI, mis esitab serverile TCP/IP protokollis päringu ning loeb vastuse.

LabView - ühenduspaneel

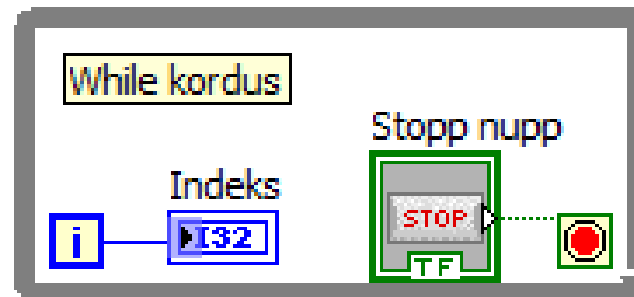
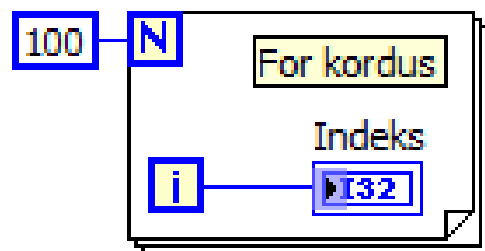
- Ühenduspaneel
 - Programmi ploki ikoon
 - Sisendite, väljundite asukohad



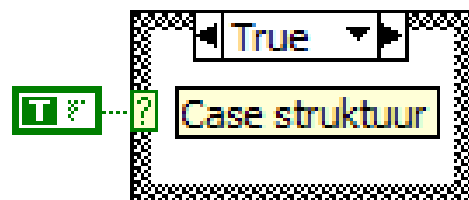
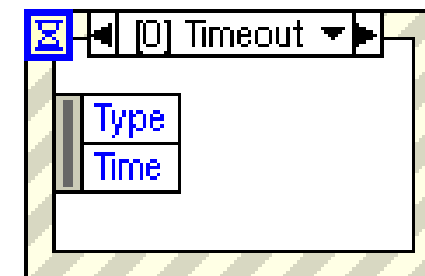
Match regular expression VI

Lihtsamad struktuurid

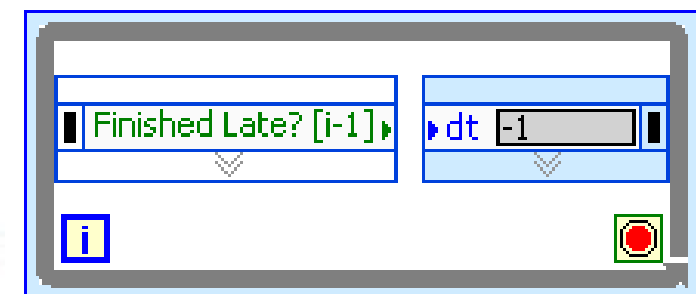
- For, while, case ja frame struktuurid
- LabView's on neist struktuuridest ka täpse ajastusega (*RealTime* paketi jaoks) variandid



Event Structure



Timed Loop

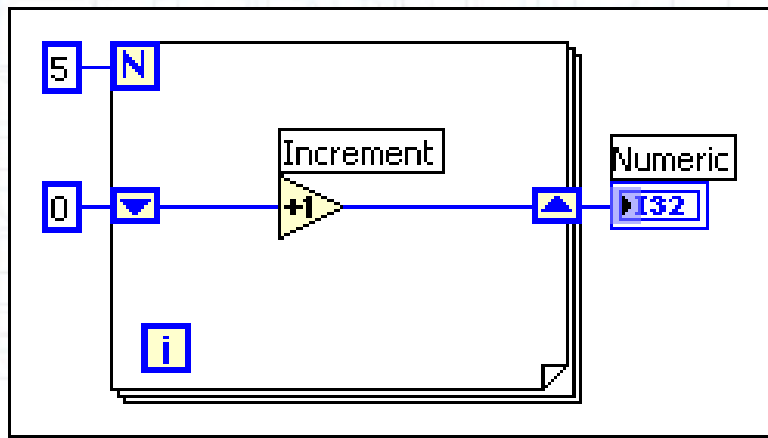


Muutujad

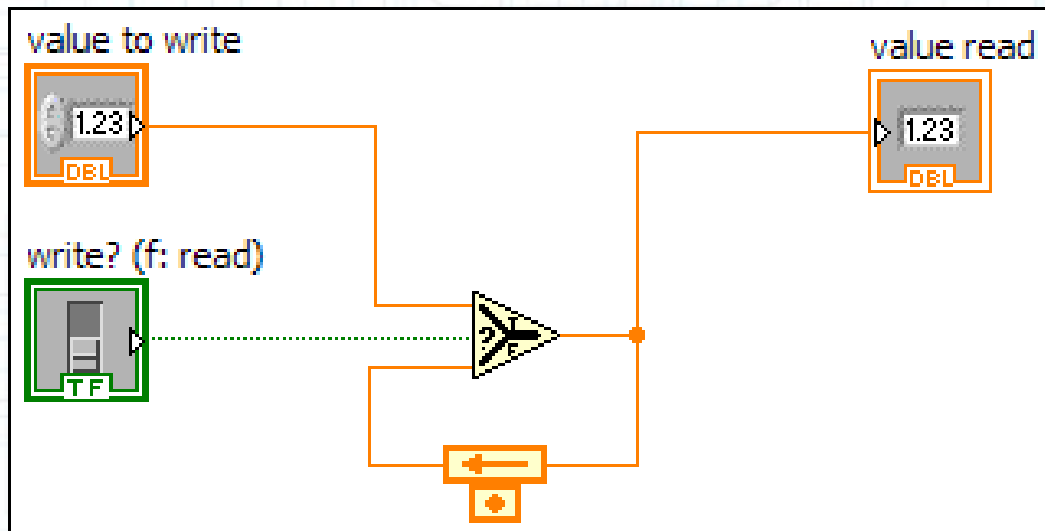
- Global – ühised mitmele VI-le
 - Ainult esipaneeliga alam-VI, mis kirjeldab ühiseid muutujaid (nagu liides mitme VI vahel)
- Local – ühised üle kogu VI
- Property Node – muutujad kontrollide / indikaatorite omaduste sättimiseks
- Üle veebi jagatud muutujad
- Kordustes kasutatakse muutujate asemel Shift-Register või Feedback Node plokke



Näide 1: shift register ja feedback node

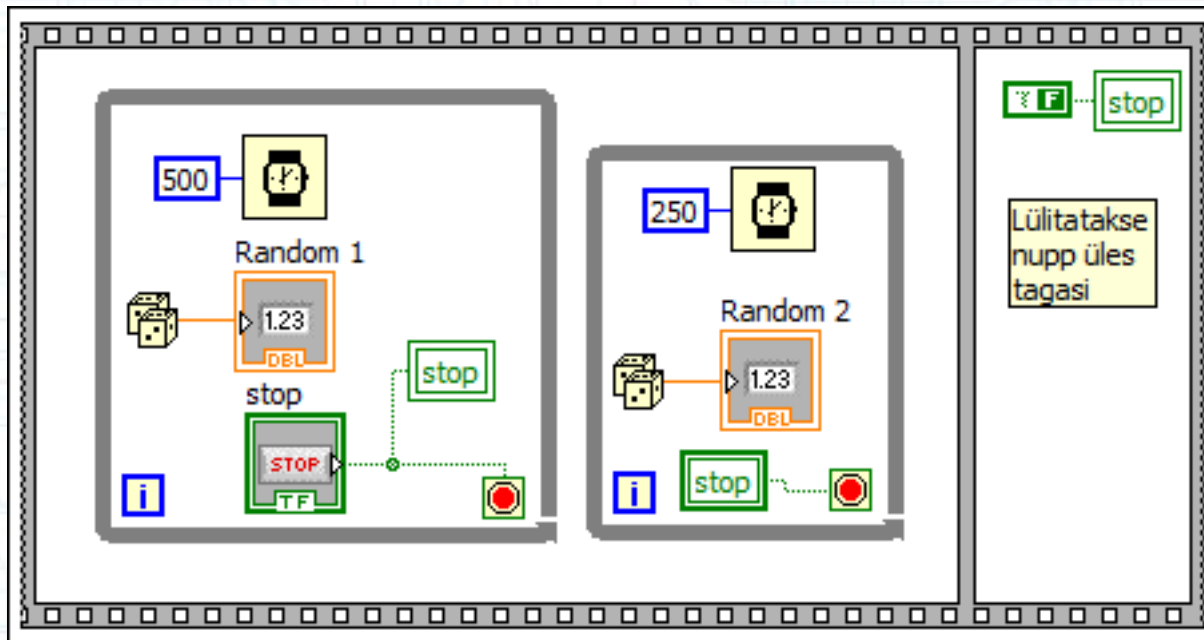


Tsükli indeks i shift-registriga



Feedback node, mida saab uuemates versioonides kasutada ka väljaspool kordusi

Näide 2: lokaalne muutuja



Random 1
0.75921

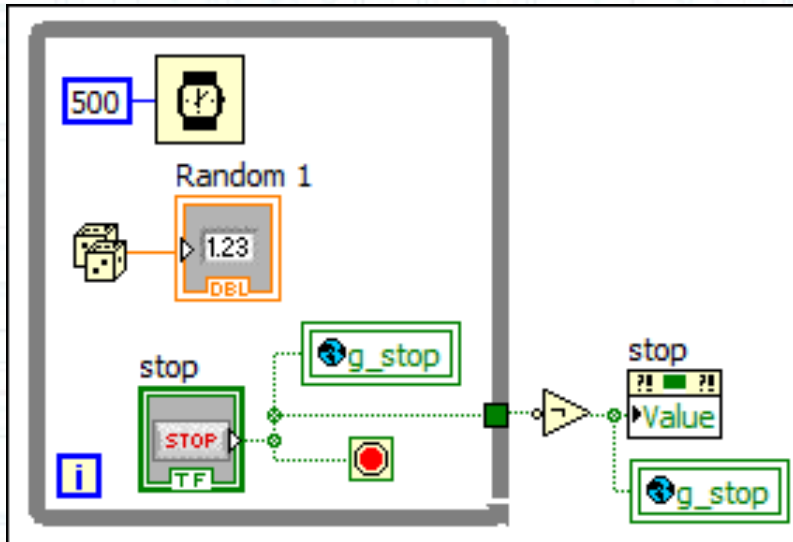
Random 2
0.30434

stop
STOP

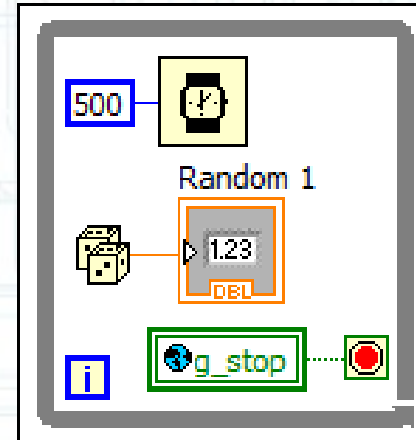
stop

Lülitatakse nupp üles tagasi

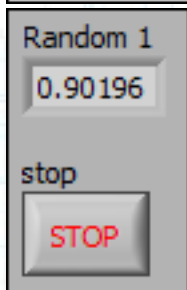
Näide 3: globaalne muutuja



Esimene VI programm



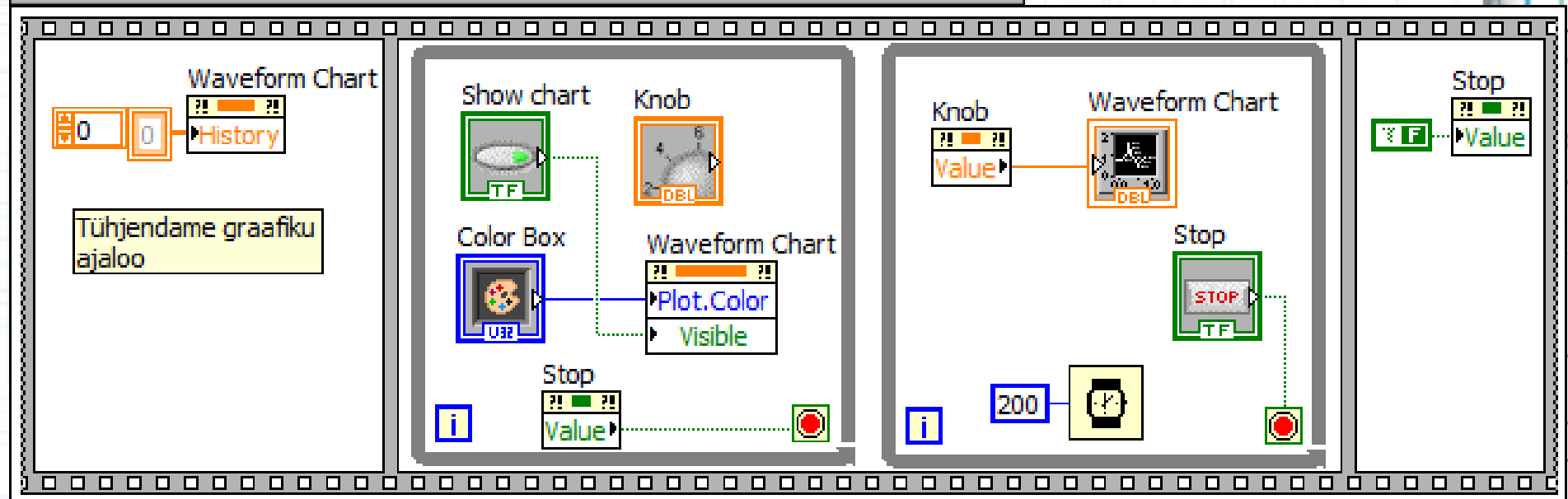
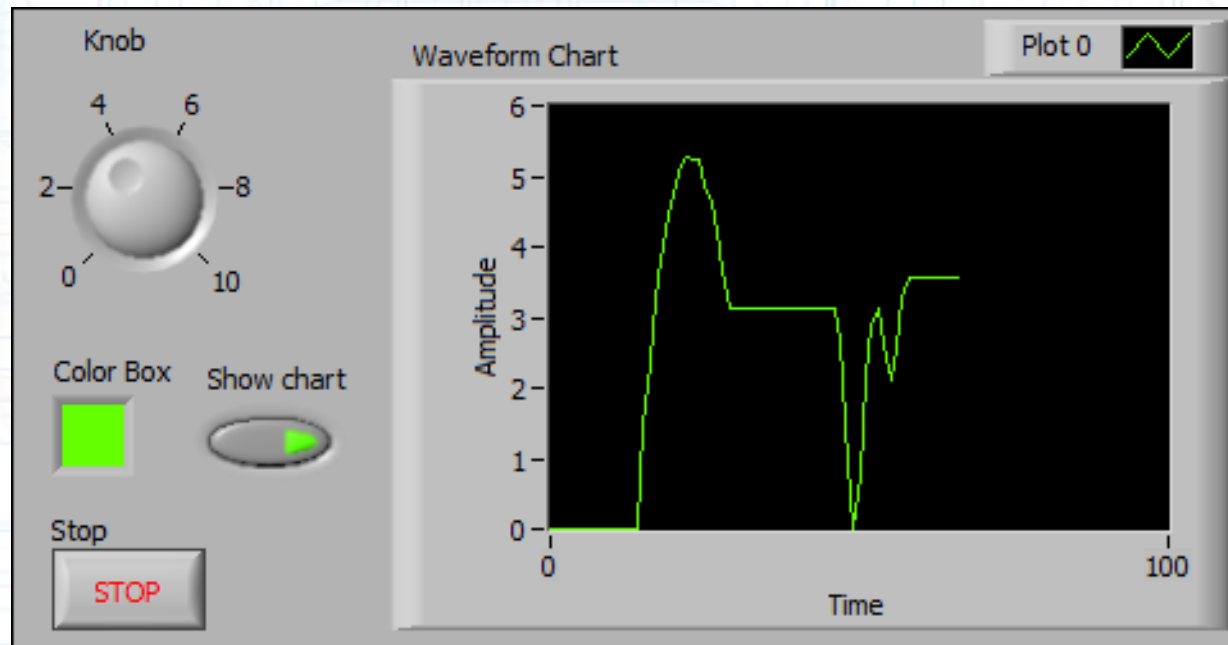
Teine VI programm



Globaalne muutuja

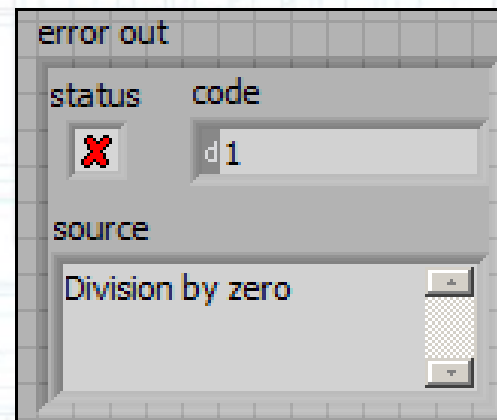
Kaks VI programmi, mis jooksevad korraga ning väljastavad rändomit, kuni esimeses programmis klikatakse STOP nupule. STOP nuppu vajutades saab eraldi VI-s defineeritud *g_stop* nimeline muutuja väärtuseks *True* ning katkestab ka teise programmi töö.

Näide 4: property node

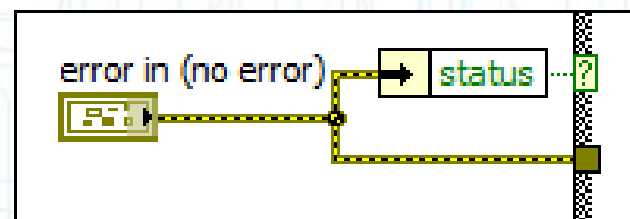


Andmete kapseldamine

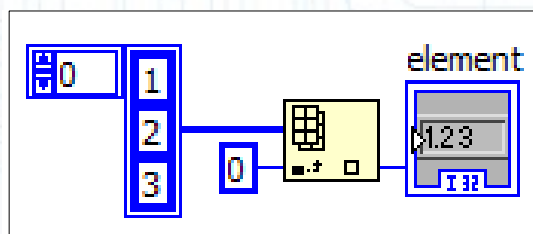
- Massiivid
- Klastrid
- Globaalsed muutujad
- Klassid



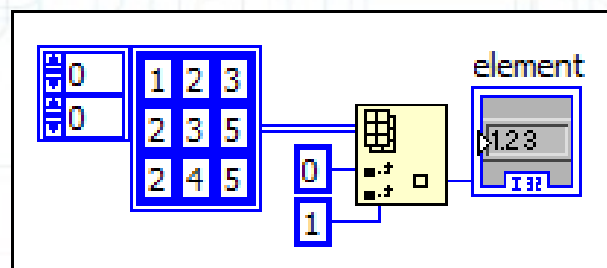
Veaklaster



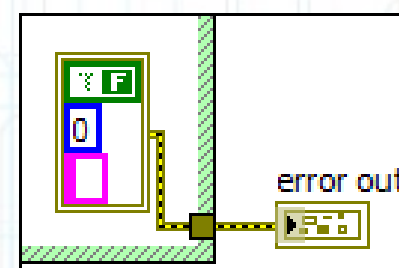
Veaklastrist staatuse väärtuse küsimine



Massiiv

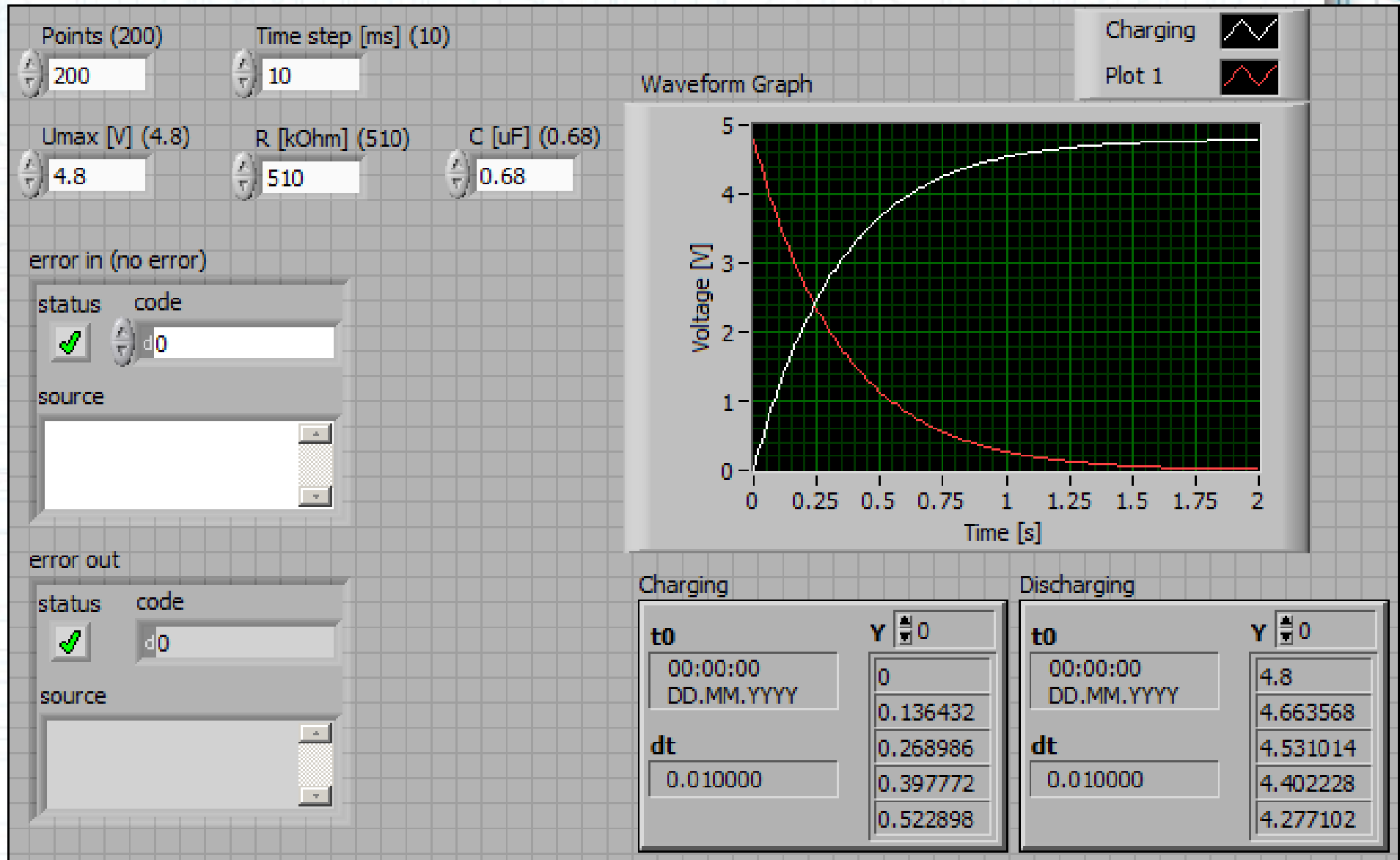


Mitmemõõtmelised massiivid

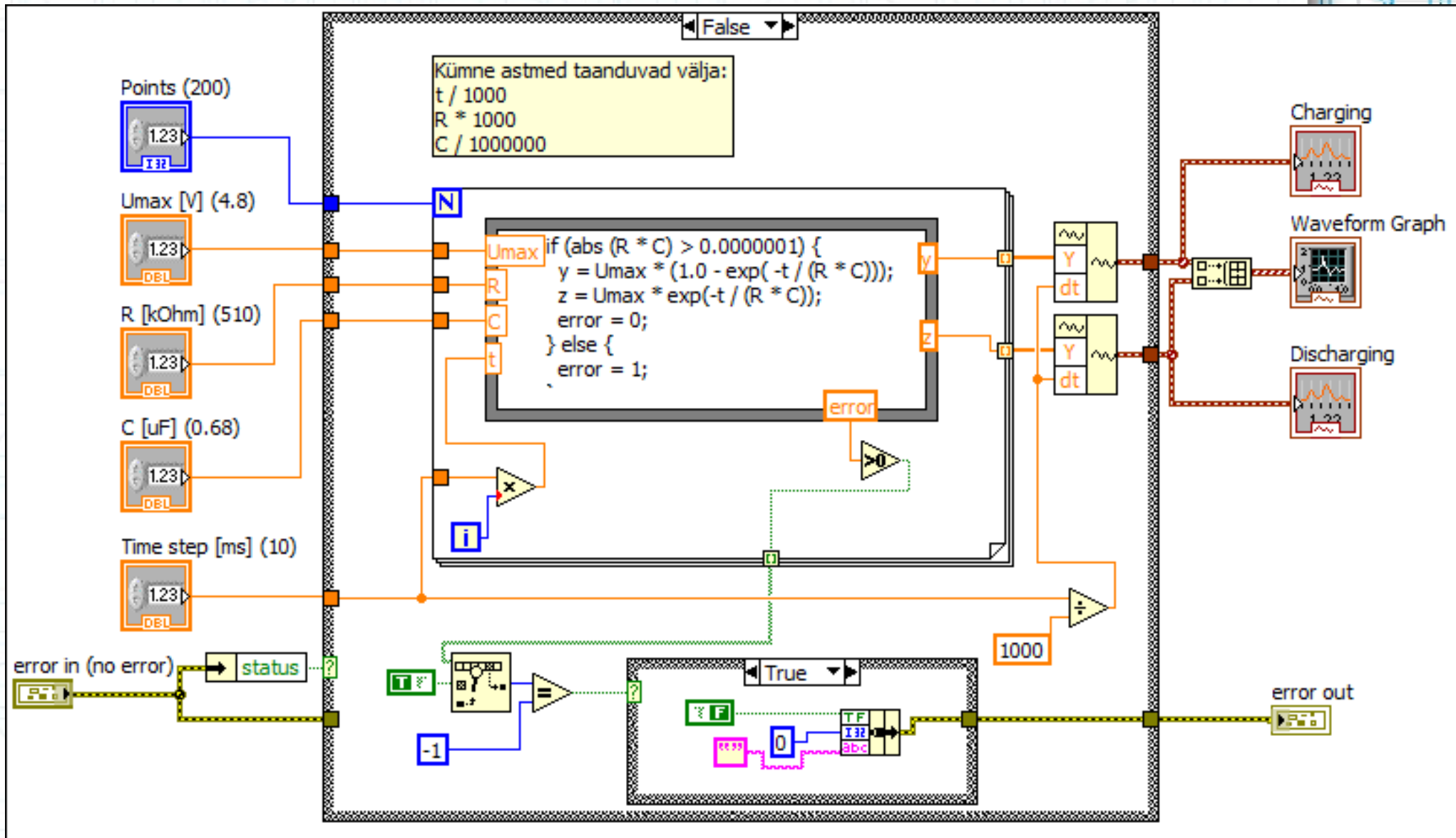


Klaster-konstandid

Näide 4, esipaneel



Näide 4, plokk-diagramm



LVOOP

- Juurklass „*LabVIEW Object*“
- Puuduvad konstruktorid
 - Andmevoo programmeerimiskeelte OOP iseärasused
- Kõik klasside liikmed on privaatsed – iga liikme lugemise / kirjutamise jaoks eraldi VI-d (*public* või *protected*)
 - Korrektsus
 - Silumine
 - Koodi sõltumatus
- *Run time Plug in*-ide süsteem
- Peaaegu sama kiire kui klastrid

Näide 5: rekursioon 0

How many times to recurse?

3

Resulting String

0 1 2 3

This demo uses recursion -- that is, a VI that calls itself -- to build a string of numbers. The task itself is trivial and could be easily achieved without using recursion, but the demo will let you see the details of recursion which might be useful for solving harder problems.

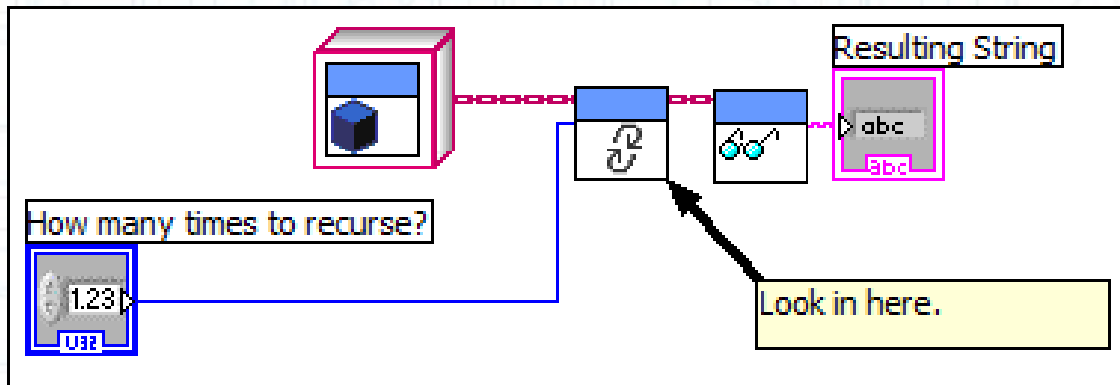
Instructions:

1. Select the number of times you want the demo to recurse.
2. Run the VI.
3. The output is a string of numbers starting at zero and going up to the input number.

Refer to the block diagram of this VI and its subVIs for more information on how recursion works.

Põhiprogrammi esipaneel

Näide 5: rekursioon 1



Põhiprogrammi plokk-diagramm

Recursion in	Recursion out
Numeric in	See block diagram.
0	
error in (no error)	error out
status code	status code
d0	d0
source	source

Klassi alam-VI rekursiivseks sõne väärtustamiseks

Recursion.lvclass in	Recursion.lvclass out
String	
error in (no error)	error out
status code	status code
d0	d0
source	source

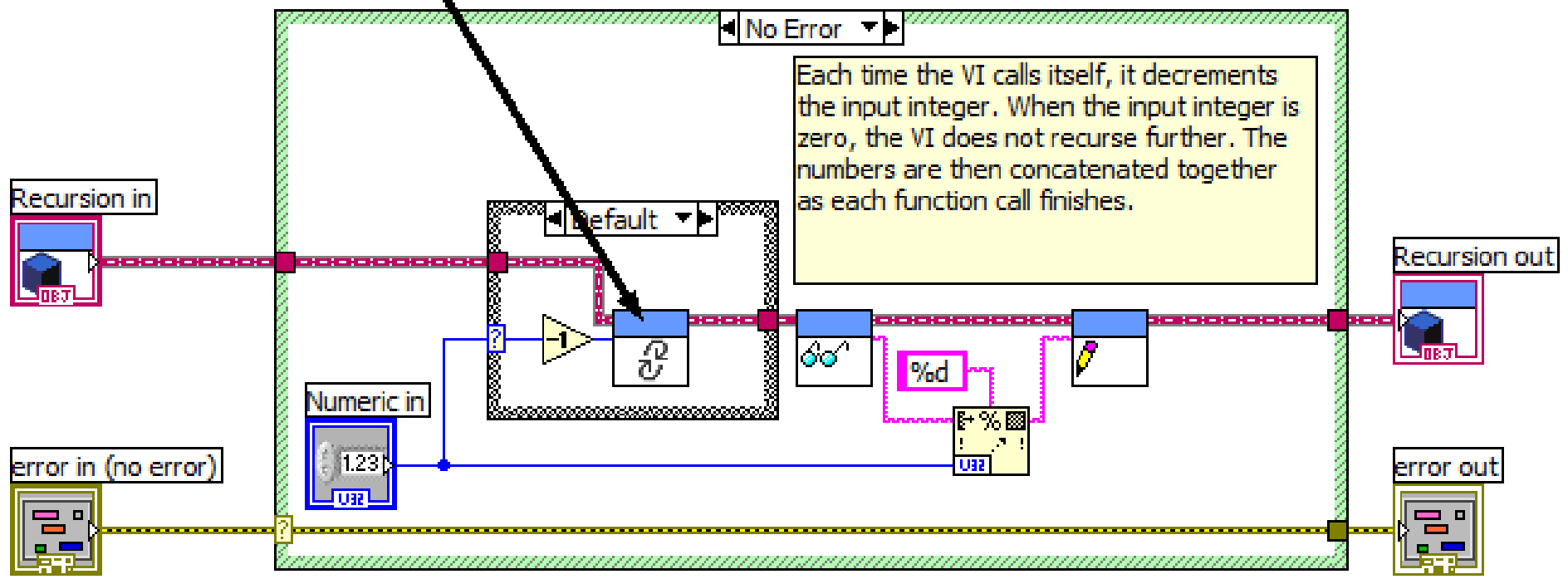
Klassi alam-VI sõne lugemiseks klassist

Näide 5: rekursioon 2

Complete the following steps to configure a VI to allow recursion:

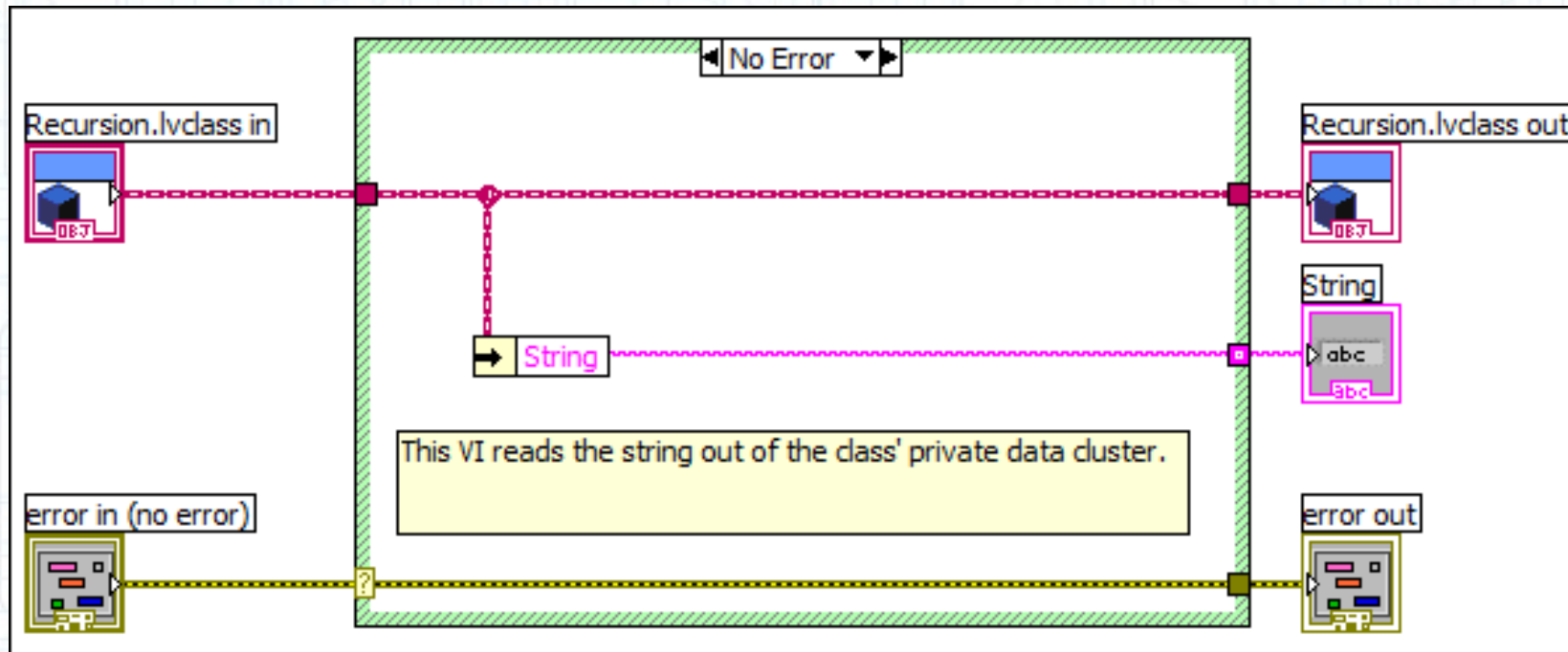
- 1) (LV 8.5 and 8.6 only. LV 2009 skip step one.) Right-click a class terminal on the connector pane of the VI and select **This Connection Is>>Dynamic Dispatch Output** from the shortcut menu. You can mark only LabVIEW class member VIs as dynamic dispatch.
- 2) Select **File>>VI Properties** and select **Execution** from the pull-down menu.
- 3) Place a checkmark in the **Reentrant execution** checkbox and select **Share clones between instances**.

This subVI invokes itself. A VI that calls itself is referred to as "recursive."



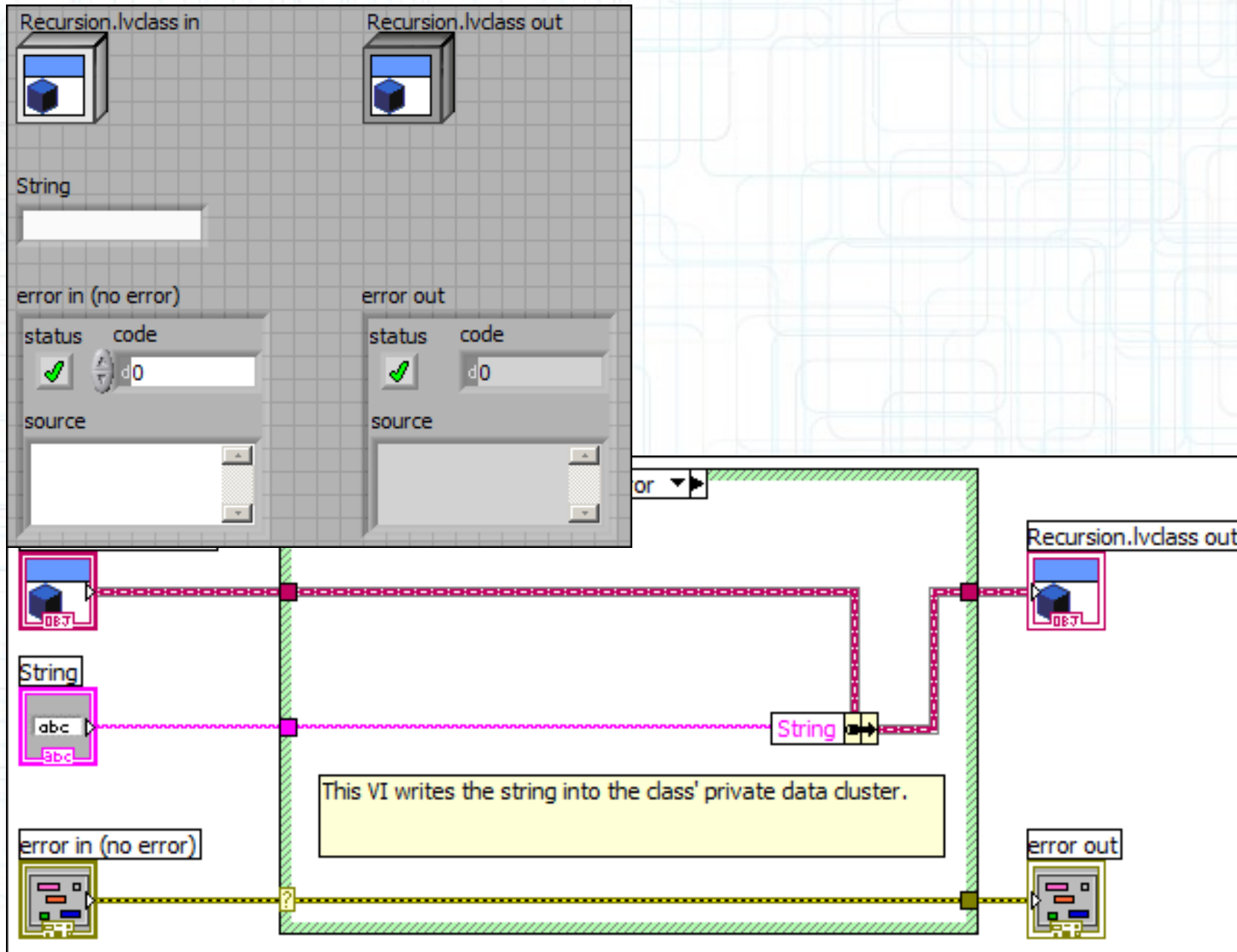
Rekursiooni alam-VI plokk-diagramm

Näide 5: rekursioon 3



Avalik alam-VI, mis loeb klassi liikmeks oleva sõne väärtuse

Näide 5: rekursioon 4



Kriitika

- Ei ole ei vaba ega open-source (paketid \$1,249 kuni \$4,699)
- Stand-alone programmid vajavad *run-time* mootorit
- Ainult windowsi masinatel korralik draiverite tugi
- Paralleelarvutusi ilma vigadeta üsna keeruline korraldada

Viited

- <http://en.wikipedia.org/wiki/LabVIEW>
- <http://www.ni.com/labview/optin/purchasing>
- <http://zone.ni.com/devzone/cda/tut/p/id/357>

Küsimusi?

