

# Programmeerimise põhikursus Javas

Loeng 8

<http://courses.cs.ttu.ee/pages/ITI0011>

# Outline

- Homework stuff
- codingbat
- **GUI, JavaFX**
- III HW
- I HW

# Homework submission

- <https://courses.cs.ttu.ee/pages/ITI0011:git>
- Homework 1 to be pushed into git before **October 20th**
  - into folder "HW1"
- Homework 2 to be pushed into git latest **October 19th 23:59**
  - into folder "HW2"
  - libraries (jar-files used) should be under lib/ (e.g. HW2/lib/twitter4j.jar)
- Homework 3 to be pushed into git latest **November 16th 23:59**
  - into folder "HW3"
  - **Android version**
- **Course code example in git:**  
<http://firstname.lastname@git.ttu.ee/kursused/iti0011/materjalid.git>
- Use UNI-ID to access materials (not visible in browser)

# Homework 2 remarks

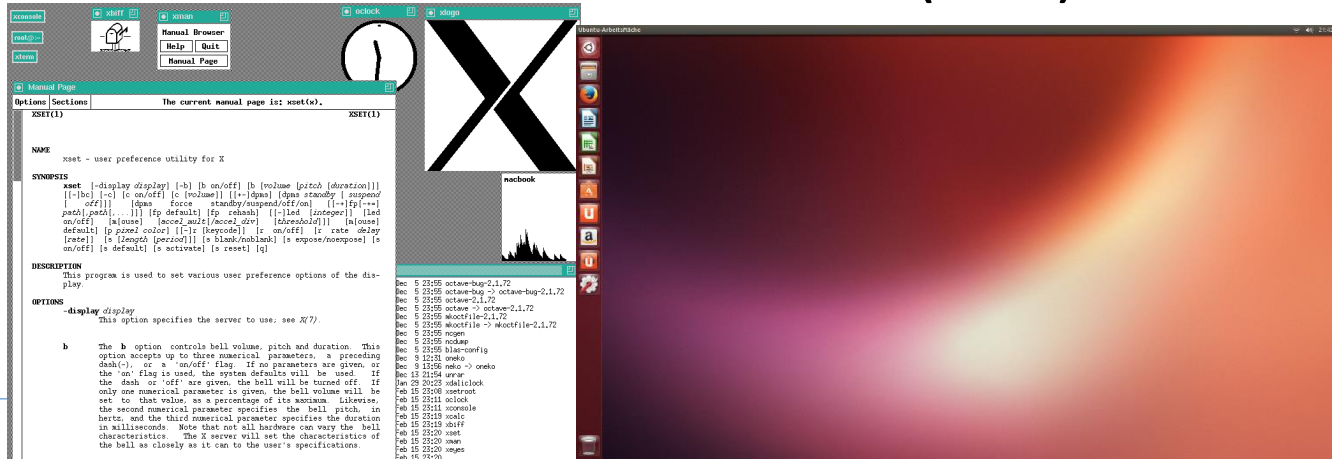
- <https://courses.cs.ttu.ee/pages/ITI0011:Säuts>
- Things to check:
  - location names with several words like "New York"
  - negative tweet count
  - non-existing locations (asfdasdfasdf)
  - locations with 0 tweets
- No need to validate cache file
  - but you should validate information in the file (negative radius etc.)

# Libraries in Eclipse

- Project > Properties > Java Build Path > Libraries
- Add external JARs..
- Browse to the jar needed

# GUI

- **Graphical User Interface (GUI)** - a type of interface that allows users to interact with devices through graphical icons and visual indicators, as opposed to text-based interfaces.
- A typical form of GUI:
  - Computer/mobile screen
  - Interaction with mouse, keyboard or by touching
- GUI should ease the steep learning curve of command-line interfaces (CLIs)



```
mars@marsmain ~ $ pwd
/home/mars
mars@marsmain ~ $ cd /usr/portage/app-shells/bash
mars@marsmain /usr/portage/app-shells/bash $ ls -al
total 130
drwxr-xr-x  3 portage portage 1024 Jul 25 10:06 .
drwxr-xr-x 33 portage portage 1024 Aug  7 22:39 ..
-rw-r--r--  1 root  root   35008 Jul 25 10:06 ChangeLog
-rw-r--r--  1 root  root   27002 Jul 25 10:06 Manifest
-rw-r--r--  1 portage portage 4645 Mar 23 21:37 bash-3.1_p17.ebuild
-rw-r--r--  1 portage portage 5977 Mar 23 21:37 bash-3.2_p39.ebuild
-rw-r--r--  1 portage portage 6151 Apr  5 14:37 bash-3.2_p48-r1.ebuild
-rw-r--r--  1 portage portage 5988 Mar 23 21:37 bash-3.2_p48.ebuild
-rw-r--r--  1 portage portage 5643 Apr  5 14:37 bash-4.0_p10-r1.ebuild
-rw-r--r--  1 portage portage 6230 Apr  5 14:37 bash-4.0_p10.ebuild
-rw-r--r--  1 portage portage 5648 Apr 14 05:52 bash-4.0_p17-r1.ebuild
-rw-r--r--  1 portage portage 5532 Apr  8 10:21 bash-4.0_p17.ebuild
-rw-r--r--  1 portage portage 5660 May 30 03:35 bash-4.0_p24.ebuild
-rw-r--r--  1 root  root   5660 Jul 25 09:43 bash-4.0_p28.ebuild
drwxr-xr-x  2 portage portage 2048 May 30 03:35 files
-rw-r--r--  1 portage portage  468 Feb  9 04:35 metadata.xml
mars@marsmain /usr/portage/app-shells/bash $ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
<herd>base-system</herd>
<use>
  <flag name='bashlogger'>Log ALL commands typed into bash; should ONLY be
  used in restricted environments such as honeypots</flag>
  <flag name='net'>Enable /dev/tcp/host/port redirection</flag>
  <flag name='plugins'>Add support for loading builtins at runtime via
  'enable'</flag>
</use>
</pkgmetadata>
mars@marsmain /usr/portage/app-shells/bash $ sudo /etc/init.d/bluetooth status
Password:
* status: started
mars@marsmain /usr/portage/app-shells/bash $ ping -q -c1 en.wikipedia.org
PING rr.esams.wikimedia.org (91.198.174.2) 56(84) bytes of data.

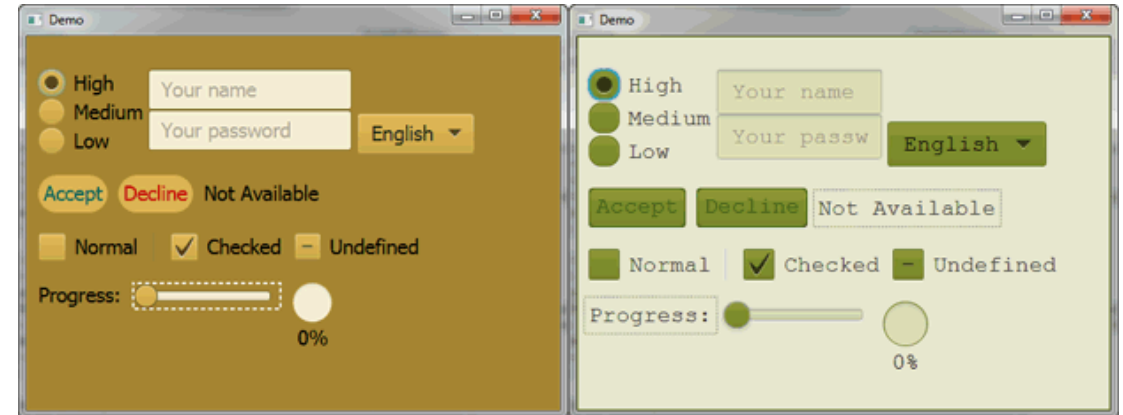
--- rr.esams.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 2ms
rtt min/avg/max/mdev = 49.820/49.820/49.820/0.000 ms
mars@marsmain /usr/portage/app-shells/bash $ grep -i /dev/sda /etc/fstab | cut --fields=3
/dev/sda1      /boot
/dev/sda2      none
/dev/sda3      /
mars@marsmain /usr/portage/app-shells/bash $ date
Sat Aug  8 02:42:24 MSD 2009
mars@marsmain /usr/portage/app-shells/bash $ lsmod
Module              Size  Used by
rndis_wlan          23424  0
rndis_host          8696   1 rndis_wlan
cdc_ether           5672   1 rndis_host
usbnet              18688  3 rndis_wlan,rndis_host,cdc_ether
parport_pc         38424  0
fglrx               2388128 20
parport             39648  1 parport_pc
iTCO_wdt            12272  0
i2c_i801            9380   0
mars@marsmain /usr/portage/app-shells/bash $
```

# JavaFX

- Software platform for creating Rich Internet Applications (RIA)
- Can be run on desktop computers and web browsers in Windows, Linux and Mac OS X
- Starting from Java version 8, JavaFX is implemented as a native Java library
  - Last version of JavaFX was 2.2
  - Current version has the same number as the Java itself: JavaFX 8
- JavaFX can also be run on Android, iOS, Raspberry Pi
- <http://docs.oracle.com/javafx/>
- JavaFX can be also used in Java 1.7, but for simplicity Java 1.8 is recommended

# JavaFX features

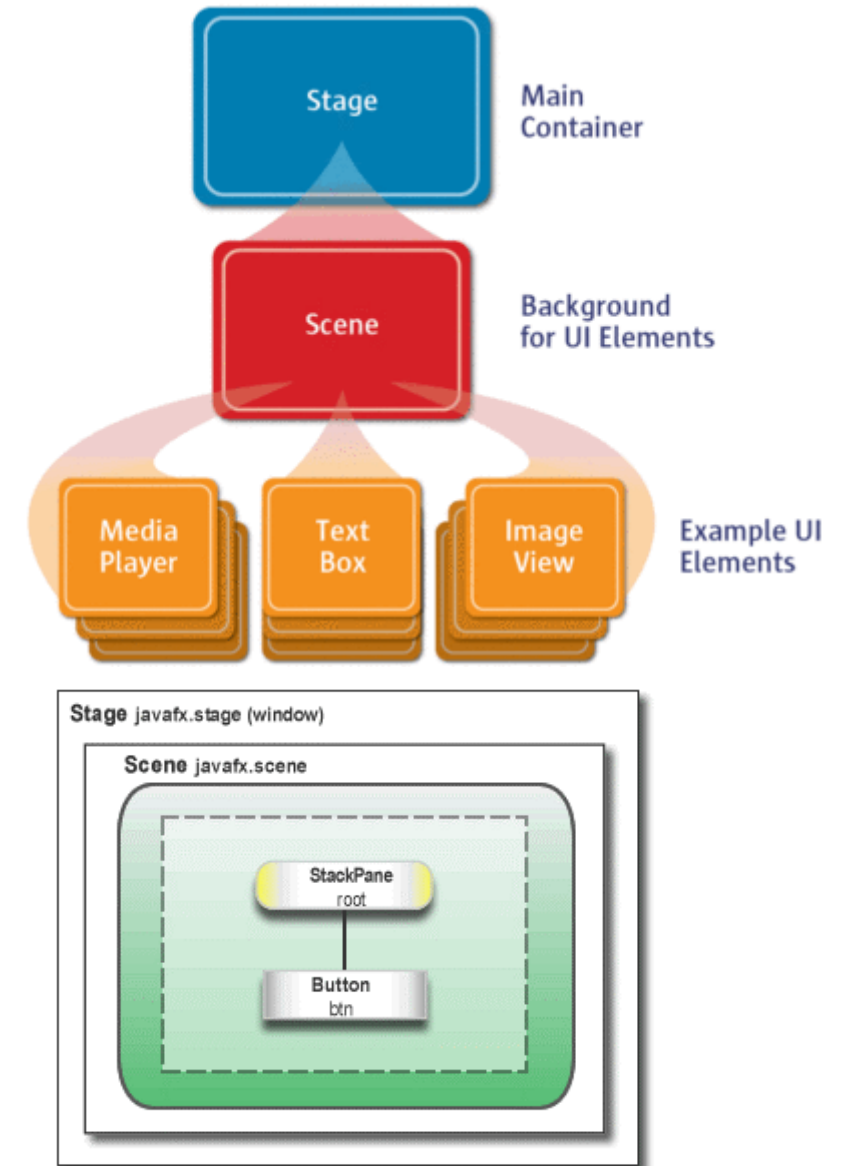
- Media and images support (visual and audio media)
- Web component (based on Webkit, full browsing functionality)
- CSS (Cascading Style Sheets) can be used for styling
- UI Controls
- Layout (different layouts to organize controls)
- 3D support
- Visual effects
- etc.





# Structure of JavaFX application

- It's like a theatre play:
  - The Stage is the main container which is usually a Window with a border and the typical minimize, maximize and close buttons.
  - Inside the Stage you add a Scene which can, of course, be switched out by another Scene.
  - Inside the Scene the actual JavaFX nodes like AnchorPane, TextBox, etc. are added.



# Everything is a node

- JavaFX Scene Graph API is responsible of rendering **GUI to the screen**
- A scene graph is a tree data structure
- Retained mode API - maintains internal model of all graphical objects
  - what objects to display
  - what areas of screen need repainting
  - how to render it all in the most efficient manner
- Instead of invoking drawing methods directly, instead use scene graph API
- Individual items held within the application are known as *nodes*

