KUClock: Code Inspection

Use Case: add clock

Lesson 09-2 v1.0

Jittat Fakcharoenphol
Fall 2007/ 2008

Goals

- To practice code reading & code inspection
- To improve the quality of the code

NOT our goals

- This is **NOT** a blame game.
  - This inspection practice session is not to blame the developers (or anyone)
  - Everyone should help them to produce better quality software

Code/design quality (1)

- Coding Guidelines/Standard
- What are code quality criteria?
  - Basic metrics [from Booch'07]:
    - Coupling 
    - Cohesion 
    - Sufficiency / Completeness
    - Primitiveness
  - Focus on classes' interfaces

Code/design quality (2)

- From HF OOA&D book
  - Basics:
    - Encapsulate what varies
    - Code to an interface
    - Class are about behavior and functionality
  - The Don't Repeat Yourself Principle (DRY)
  - The Single Responsibility Principle (SRP)
  - The Open-Closed Principle (OCP)
  - The Liskov Substitution Principle (LSP)

Code/design quality (3)

- For a list of more heuristics, see, e.g., http://lcm.csa.iisc.ernet.in/soft_arch/OO_Design_Heuristic.htm

Basic architecture of GUI apps

- Model-View-Control:
  - **Model** represents domain-specific information
  - **View** displays the model, allows interaction
  - **Controller** processes and responses to events
- MVC in Java:
  - class Observable, interface Observer
    - see: http://java.sun.com/javase/6/docs/api/java/util/Observable.html

Review: Event Handling in AWT

- You create a listener object that implements some interface and assign it to the ui component (usually with an anonymous class).
- E.g.,

```
clockMenu.addMouseListener(new MouseListener()
{
   public void mouseEntered(MouseEvent arg0) {
      if(ctrl.getClockPanel() == null ){
         ...
      } else {
         ...
      }
   }
});
```

Use-case-driven inspection

- Consider each use case
- Use interaction diagram to trace the code
  - If we do not have any interaction diagram, we will build it along the way

Use case: add clock

Scope KasetClock
Level User-Goal
Primary Actor User
Stakeholders and Interests User : Add more clock then the clock that user selection will show on the main page.

Main Success Scenario
1.The user choose add clock menu
2.System creates new clock.
3.System show all of the cities name list.
4.User select the time zone(GMT) or the city.
5.System show the new clock of the city that user choose.
6.System show that city name.

Hints for Inspecting KUClock (1)

- Main is in `ui/Main.java`
- Main JFrame is in `ui/ClockUI.java`
- Controller is in `model/Controller.java`

Hints for Inspecting KUClock (2)

- Add clock:
  - by menu:
    - ClockUI.MenuListener.actionPerformed
  - by mouse click:
    - ClockPanel.mouseClicked
    - ClockPanel shows PopupMenu (created by Controller)
    - Controller.popupmenu calls actionListener for addMenu (defined in Controller, line 24)

Output from the inspection

- Form a group of 2-3 people
- Inspect KUClock's code by tracing the execution of use case “add clock.”
- Each group presents 2-3 specific comments/suggestions, regarding, e.g.,
  - Defects
  - Design quality (from code)
  - Code quality
- Your comments/suggestions **must** be accompanied with the code samples

Example: hard-coded user guide

- **Line:** 316 – 359 in ClockUI.java
- **Code:**

```
else if (menu.getActionCommand().equals("User Guide")) {
    JScrollPane pane = new JScrollPane();
    JOptionPane.showMessageDialog(pane,
            "KUCLock\n\n" +
            "This product is created by student who participate in\n" +
            "219343 Software Validation Verification and Testing 1st\n" +
            "on purpose of studying software project.\n\n" +

            "How to use KUClock\n" +
            "1. Add Clock\n" +
            ... (~30 more lines) ...
            "   set the clock to 24hours system or the Am-Pm system too.",
            "User Guide", JoptionPane.INFORMATION_MESSAGE);
}
```

- **Suggestion:** should be placed in an external data file
- **Reason:** maintainability, better support for i18n