| | Design patterns |
|---|---|
| Lecture 5 Topics on Refactoring Some materials are based on Martin Fowler's book Copyright © Yijun Yu, 2005 | We showed the structures of all the classic design patterns We explained some of them and their applications in OpenOME, Protégé and Eclipse For your exercise Explain the MVC and plug-in patterns using the classic design patterns Find more cases in OpenOME that can apply the design patterns |
| Spring 2005 ECE450H1S Software Engineering II | The application of design patterns, can be called "refactoring" Spring 2005 ECE450H1S Software Engineering II |
| Today Topics on Refactorings 1. What is refactoring? Why? 2. How to classify refactorings 3. How to apply refactorings 4. Compare with tuning and design patterns 5. Refactor source code into requirements 6. Summary | What is refactoring? It is a new English word, can be used in part of speech for a noun (countable or uncountable), a verb Its origin = Factoring |
| References Martin Fowler. Refactoring – improve the design of existing code. <u>http://www.refactoring.com</u> Tom Mens et al. "A survey of software refactoring". TSE 30(2), 2004. Spring 2005 ECE450H1S Software Engineering II | Spring 2005 ECE450H1S Software Engineering II |

Last lecture and tutorial ...

Factoring

- In mathematics, factorization or factoring is the decomposition of an object into an expression of smaller objects, or factors, which multiplied together give the original
- For example, the number 15 factors into primes as 3×5 ; and the polynomial x^2 - 4 factors as (x - 2)(x + 2)

Refactoring

- **Refactoring** is the process of rewriting written material to improve its *readability* or structure, with the explicit purpose of keeping its meaning or behavior.
 - The term is by analogy with the *factorization* of numbers and polynomials. For example, x² 1 can be factored as (x + 1)(x 1), revealing an internal structure that was previously not visible (such as the two zeroes at +1 and -1). Similarly, in software refactoring, the change in visible structure can often reveal the "hidden" internal structure of the original code.
- Extracting common descriptions 20 + 20 + 20 = (1 + 1 + 1) x 20 = 3 x 20

| Spring 2005 | ECE450H1S | Software Engineering II | Spring 2005 | ECE450H1S | Software Engineering II |
|-------------|-----------|-------------------------|-------------|-----------|-------------------------|
| | | | | | |

Software Refactoring

- Software refactoring = "Restructuring existing code by altering its internal structure without changing its external behavior"
 - adapted from Martin Fowler's book

• To avoid duplications

A. Hunt, and D. Thomas. *Pragmatic Programmer*, Addison Wesley,1999. Martin Fowler, *Avoid Repetition*, IEEE Software, Jan/Feb 2001 pp.97—99.

More on definitions

Are the following activities refactorings?

- Adding new functionalities
- Fixing correctness bugs
- Tuning performance
- Patching security holes

When to apply refactorings The refactoring rhythms "Any fool can write code that a computer can Development = understand. Good programmers write code that human (Adding features, Refactoring)* can understand" Bad code smells: Refactoring = (Testing, Small Steps) * Duplicate code (clones): feature envy • Small Steps = one of the refactoring types Complex control, Long method use Hammock graph: single entry/single exit Comments signal semantic distance - Conditional and loops Complex data, Long parameter list OO specific: large class, switch statements, parallel ٠ inheritance, middle man, message change, temporary fields, data class, etc. Spring 2005 ECE450H1S Software Engineering II Spring 2005 ECE450H1S Software Engineering II

2. Type of refactorings

"Putting things together when changes are together"

- Extract Methods
- Move Methods
- Rename Methods
- Replace Temp with Query
- Replace conditionals with polymorphism
- Replace Type code with State/Strategy
- Self Encapsulate Field

•

Spring 2005

ECE450H1S Soft

Software Engineering II

3. Applications

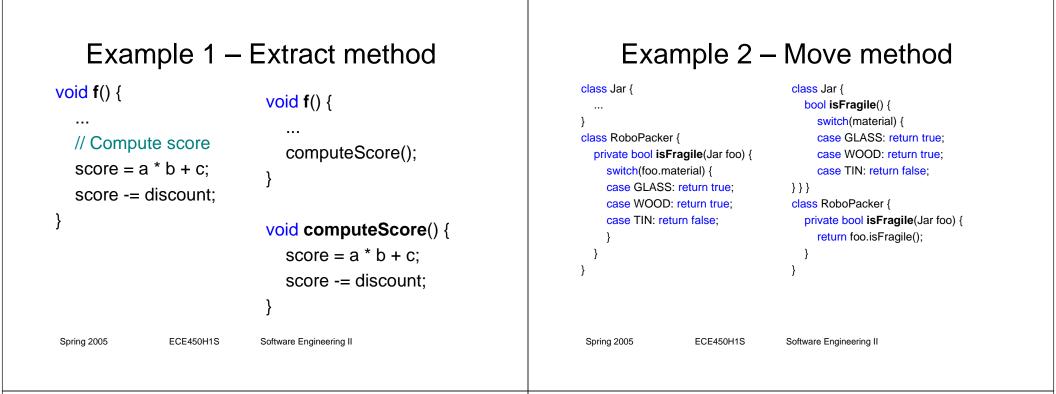
- We use three examples to explain some basic refactorings
 - Extract method:
 - signalled by comments
 - single-entry, single-exit
 - · increase the level of indirection
 - reduce the length of a method

ECE450H1S

- increase the chance of reuse
- Move method:
 - Place method together with the object, *Putting things* together when changes are together
- Replace conditions with polymorphism
 - Switches are "hard code", polymorphism is better for extensibility in OO

| Spring | 2005 |
|--------|------|
| | |

Software Engineering II



Example 3 – Replace conditionals with polymorphism

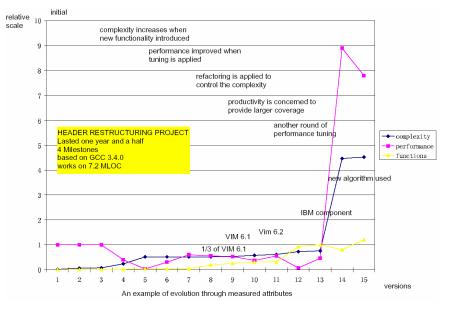
| class Jar { | class Jar { |
|-------------------------------|---|
| <pre>bool isFragile() {</pre> | <pre>bool isFragile() {</pre> |
| switch(material) { | return material.isFragile(); |
| case GLASS: | } } |
| // complex glass calculation | |
| case WOOD: | interface Material { } |
| // complex wood calculation | <pre>class GlassMaterial:Material { }</pre> |
| case TIN: | <pre>class WoodMaterial:Material { }</pre> |
| // complex tin calculation | <pre>class TinMaterial:Material { }</pre> |
| }}} | |
| | |

4. Refactoring versus Tuning

- Refactoring aims at improve understandability and maintainability
- Tuning aims at improve performance
- They are both non-functional (no new features), but they are different
 - Refactoring can be harmful to performance
 - Tuning can be harmful to maintainability
- · You need to know where are the bottlenecks
- Y. Yu et al. "Software refactorings guided by softgoals", *REFACE workshop in conjunction with WCRE'03*.

Spring 2005

The header restructuring project



Your exercise

- Monitor the evolution of your software product by measured its metrics
 - Statically: complexity metrics: LOC, Halstead, McCabe
 - Dynamically:

Performance metrics: time (clockticks, #instructions), space (cache misses, L1 instruction, L1 data, L2 cache, etc., memory footprint)

 Decide on which is the urgent nonfunctional task

ECE450H1S

Spring 2005

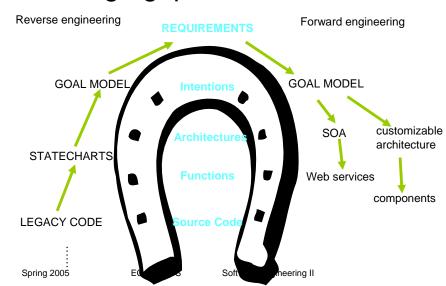
Software Engineering II

5. Refactoring into Requirements

Motivation to recover requirements from source code

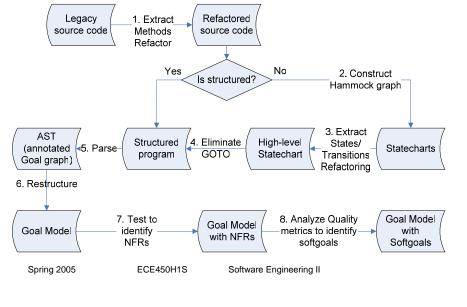
- Requirements are lost in documentations, sadly, it is very common in the software development practices
- Legacy software code are not explained in documentation
- Mismatch between implementations and requirements

Spring 2005 ECE450H1S Software Engineering II



Huge gap in abstractions

A semi-automatic process



Example. Columba Refactoring

- Search "Java email client" in Google, you will find this software
- It is open-source
- It has 140 KLOC in Java
- It also has plug-in patterns

ECE450H1S

 First thing, we modify the code base to fit Eclipse development (moving packages, i.e., move all "src" subdirectories including plug-in projects under the same "src" directory)

Spring 2005

Software Engineering II

A screenshot

| Seriel Rems (total: 0) Coal Folders Coal Folders Dob Recent Messages | 🖾 👔 💡 🗑 🖲 Subject or Sender go | | | |
|---|--------------------------------|------|----------------------|--------------|
| E fib Inbox | | | 1 Contraction of the | read:0 unre |
| | D B F / Subject | Prom | Date Size | Search dialo |
| Learn messages Loging messages Charge mes Charge mes Drafs Drafs Drafs Drafs Sent Sent Sent Sent Sent Sent Sent | | | | |
| 🗄 - 🕥 Yijun Yu | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Where to look at first?

• Secondly, we look for the main routine from the manifest in the JAR file

Manifest-Version: 1.0 Ant-Version: Apache Ant 1.6.2 Created-By: 1.4.2_06-b03 (Sun Microsystems Inc.) Main-Class: org.columba.core.main.Main Sealed: false Class-Path: lib/usermanual.jar lib/junit.jar lib/lucene-1.3-final.jar

lib/commons-cli-1.0.jar lib/jwizz-0.1.2.jar lib/plastic-1.2.0.jar li b/jhall.jar lib/forms-1.0.4.jar lib/ristretto-1.0_RC2.jar lib/jscf-0. 2.jar lib/macchiato-1.0pre1.jar lib/frapuccino-1.0pre1.jar lib/winpac k.jar lib/jniwrap-2.4.jar lib/jdom.jar lib/jpim.jar lib/je.jar \${lib. jdic}

Spring 2005

The Main routine

public static void main(String[] args) {
 Main.getInstance().run(args);

Thus we look at "run" routine, which has 81 lines of code

Spring 2005

Software Engineering II

The Run routine

| 1 ColumbaLogger.createDefaultHandler(); 2 registerCommandLineArguments(); 3 // handle commandLineArguments(); 4 if (handleCoreCommandLineParameters(args)) { 5 System.exit(0); 6 } 7 // prompt user for profile 8 Profile profile = ProfileManager.getInstance().getProfile(path); 9 // initialize configuration with selected profile 10 new Config(profile.getLocation()); 11 // if user doesn't overwrite logger settings with commandline arguments 12 // just initialize default logging 13 ColumbaLogger.createDefaultHandler(); 14 ColumbaLogger.createDefaultFileHandler(); 15 ColumbaLogger.createDefaultFileHandler(); 16 17 for (int i=0; i <args.length; i++)="" td="" {<=""> 18 LOG.info("arg["+i+"]="+args[I]); 19 } 20 </args.length;> | pub | lic void run(String args |]) { | |
|---|---------------|--------------------------|-----------|-------------------------|
| <pre>3 // handle commandLine parameters 4 if (handleCoreCommandLineParameters(args)) { 5 System.exit(0); 5 // prompt user for profile 8 Profile profile = ProfileManager.getInstance().getProfile(path); 9 // initialize configuration with selected profile 1 new Config(profile.getLocation()); 11 // if user doesn't overwrite logger settings with commandLine arguments 12 // just initialize default logging 13 14 ColumbaLogger.createDefaultHandler(); 15 ColumbaLogger.createDefaultFileHandler(); 16 17 for (int i=0; i<args.length; 18="" arg["+i+"<="" arg["+i+"]="+args[i]); 19 } 20</pre></td><td>16</td><td></td><td></td><td></td></tr><tr><td>19 } 20</td><td>17</td><td>for (int i=0; i<args.ler</td><td>ngth; i++) {</td><td></td></tr><tr><td>20</td><td>18</td><td>LOG.info(" i++)="" log.info("arg["+i+"]="+args[i]); 19 } 20</pre></td><td>1</td><td>ColumbaLogger.create</td><td>eDefaultHandler();</td><td></td></tr><tr><td> if (handleCoreCommandLineParameters(args)) { System.exit(0); } / prompt user for profile Profile profile = ProfileManager.getInstance().getProfile(path); // initialize configuration with selected profile new Config(profile.getLocation()); // if user doesn't overwrite logger settings with commandline arguments // just initialize default logging ColumbaLogger.createDefaultHandler(); ColumbaLogger.createDefaultFileHandler(); for (int i=0; i<args.length; i++) { LOG.info(" td="" {=""><td>]="+args[i]);</td><td></td></args.length;></pre> |]="+args[i]); | | | |
| | 19 | } | | |
| Spring 2005 ECE450H1S Software Engineering II | 20 | | | |
| Spring 2005 ECE450H1S Software Engineering II | | | | |
| Spring 2005 ECE450H1S Software Engineering II | | | | |
| Spring 2005 ECE450H1S Software Engineering II | | | | |
| | Spri | ng 2005 | ECE450H1S | Software Engineering II |
| | | | | |
| | | | | |

The Run routine refactored

public void run(String args[]) {

ColumbaLogger.createDefaultHandler(): registerCommandLineArguments(); ComponentPluginHandler handler = register_plugins(); handler.registerCommandLineArguments(); handle_commandline_parameters(args); Profile profile = prompt_user_for_profile(); initialize_configuration_with_selected_profile(profile); initialize_default_logging(args); SessionController.passToRunningSessionAndExit(args); enable_debugging_repaint_manager_for_swing_gui_access(); StartUpFrame frame = show_splash_screen(); register protocol handler(): load_user_customized_language_pack(); initialize_plugins(handler); load_plugins(); set_look_and_feel(); init_font_configurations(); set_application_wide_font(); hide splash screen(frame): handle commandline arguments of the modules(handler); restore frames of last session(); ensure native libraries initialized(): post startup of the modules(handler);

ECE450H1S

Identify NFR and introducing softgoals

public boolean usability_language_customization = false; public boolean usability_assured_progress = false; public boolean usability_look_and_feel = false; public boolean usability_font_configuration = false; public boolean extensibility = false; public boolean maintainability_debugging = false; public boolean maintainability_logging = false; public void run(String args[]) { if (maintainability_logging) ColumbaLogger.createDefaultHandler(); registerCommandLineArguments(); ComponentPluginHandler handler = register_plugins(); handler.registerCommandLineArguments(); if (extensibility) handle_commandline_parameters(args); Profile profile = prompt_user_for_profile(); initialize_configuration_with_selected_profile(profile); if (maintainability logging) initialize default logging(args); SessionController.passToRunningSessionAndExit(args); if (maintainability_debugging) enable_debugging_repaint_manager_for_swing_gui_access(); StartUpFrame frame = null; if (usability_assured_progress) { frame = show_splash_screen(); } register_protocol_handler(); if (usability_language_customization) load_user_customized_language_pack() initialize_plugins(handler); if (extensibility) load_plugins(); if (usability_look_and_feel) set_look_and_feel(); init_font_configurations(); if (usability_font_configuration) set_application_wide_font(); if (usability_assured_progress) hide_splash_screen(frame); if (extensibility) handle_commandline_arguments_of_the_modules(handler); restore_frames_of_last_session(); if (extensibility) ensure native libraries initialized(); if (extensibility) post_startup_of_the_modules(handler); Spring 2005 ECE450H1S Software Engineering II

Spring 2005

The system without the NFRs

| iva - Main, java - Eclipse Platform | | 🖾 Columba | | |
|---|--|-------------------------------|--|--|
| Edit Source Refactor Navigate Search Pr | | File Edit View Folder Message | Utilities Help | |
| • 🗟 🖻 🛔 • 🎒 🏷 • Q • Q | • 🛛 🗑 🎯 • 🎯 🛷 🌙 | - | | |
| exkage Explorer X Herardry | Main. java 23 | New Message 🛛 Receive. | iendi 💭 Repty 🤤 Forward 📓 🤤 🦉 🔘 | |
| | Man. java 23 | 2010-00 C | | |
| 0 0 @ E 😫 💌 | | P A Local Folders | | |
| * 倍 16 | /* Yijun Yu: Softg | | 🖾 😨 💡 🗃 🐌 Subject or Sender contains: | Reset Search dialog |
| + F2, Ib.Ub | public boolean usa | P 😰 Inbox | | A Design of the second state of the second sta |
| + E lb.ub.encodings | public boolean usa | Recent Messages | | |
| + FE Ib-Ub.pawt | public boolean usa | Subject contains Colun | | |
| + 由 Ib.Ub.test | public boolean usa | - Change mer | | |
| + org.columba.addressbook.config | public boolean usa | La Diana | | |
| + erg.columba.addressbook.facade | public boolean exte | | | |
| + e org.columba.addressbook.folder | public boolean main | | | |
| + erg. columba.addressbook.folder.i | public boolean main | | | |
| 🗧 🗰 org.columba.addressbook.gui | public boolean mail | Search Results | | |
| + 🖶 org.columba.addressbook.gui.acti | - public void run(St | | | |
| 🐑 🌐 org.columba.addressbook.gui.aut | | - U tijun tu | | |
| 🔹 🌐 org.columba.addressbook.gui.dial | if (maintainab) | 1 | | |
| 🐑 🌐 org.columba.addressbook.gui.dal | registerCompany | | | |
| 🔹 🌐 org.columba.addressbook.gui.dial | ComponentPlugi | | | |
| 🐑 🌐 org.columba.addressbook.gui.frai | handler.regist | | | |
| 🔹 🎂 org. columba. addressbook. gui. list | if (extensibil) | | | |
| 🐑 🌐 org.columba.addressbook.gui.tab | Profile profile | | | |
| 🔹 🌐 org.columba.addressbook.gui.tab | initialize_con: | 1 1 | | |
| 🐑 🖶 org.columba.addressbook.gui.tab | if (maintainab | • | | |
| + 🖶 org.columba.addressbook.gui.too | SessionControl. | • | | |
| + 🌐 org.columba.addressbook.gui.trei | if (maintainab) | • | | |
| 🕷 🎂 org.columba.addrezsbook.gui.tree | Bei | | | |
| + 🖶 org.columba.addressbook.gui.uti | if (usability_ | 1 1 | | |
| 🖲 🌐 org.columba.addressbook.main | register_proto | | | |
| + 🖶 org.columba.addressbook.model | if (usability_ | • | | |
| B org.columba.addressbook.parser B org.columba.addressbook.plugin | initialize_plu | | | |
| Grg.columba.addressbook.shutdo | if (extensibil) | • | | |
| | if (usability_ | • • | | |
| + erg.columba.chat | init_font_conf | | | |
| + = org.columba.chat.command | | | | |
| + erg.columba.chat.config | | | | |
| + e org.columba.chat.frame | Problems Javadoc Declaration | | | |
| + erg.columba.chat.jabber | New_configuration [Java Application] C:\Pr | | | |
| + erg.columba.chet.ui.action | Feb 1, 2005 9:55:24 AB org | | | |
| + erg.columba.chat.ul.conversation | INFO: initialising plugin- | | | |
| 🐮 🖶 org.columba.chat.ui.conversation | Feb 1, 2005 9:55:24 AB org | | | |
| 🐮 🌐 org.columba.chat.ui.conversation | INFO: initialising plugin- | • | | |
| 🐮 🌐 org.columba.chat.ui.dialog | Feb 1, 2005 9:55:24 AR org | | | < |
| 🔹 🌐 org.columba.chat.ui.presence | INFOr initialising plugin- | | | |
| 🖲 🌐 org.columba.chat.ui.roaster | Feb 1, 2005 9:55:24 AR ong | .columba.core.plugin.Mistre | ctFluginHandler (init) | |
| erg.columba.core.action | | handler: org.columbs.mail.t | | |
| 🐮 🌐 org.columba.core.backgroundtask | | .columbe.core.plugin.Abstre | | |
| 🐮 🌐 org.columba.core.charset | | handler: org.columbs.mail.m | | |
| + 🖶 org.columba.core.command | | . columba.core.plugin.Abrira | | |
| 🐮 🌐 org.columba.core.config | | bandler: org.columba.mail.c | | |
| * 🖶 org.columba.core.externaltools | | .columbe.core.plugin.distre | | |
| 🛞 🌐 org.columba.core.facade | | handler: org.columbe.meil.p | | |
| 🔹 🌐 org.columba.core.gui | | .columbs.mail.gui.message.v | | |
| 🐮 🌐 org.columba.core.gui.action | | | .jar//org/columba/core/images/ | |
| * 🖶 org. columba.core.gui.config * 🖶 org. columba.core.gui.externatico 🌱 | and a standard and a standard and a standard and a | Contractor of the contractor | The state of the second s | |
| an and contrained cover day excernation | | | | |

6. Summary

- The concepts of refactoring
- The relation to restructuring, reengineering, design patterns, performance tuning, and requirements are explained
- Refactoring is not limited to OO software, that's the major different from the design patterns
- Refactoring is not aiming at all quality attributes, they are mainly for maintenance
- Refactoring is used to reveals new structures, thus it can be used to increase the level of abstraction gradually, leading to even requirements
- A lot research is coming ...

ECE450H1S

Spring 2005

Software Engineering II

Further readings

- Martin Fowler. *Refactoring improve the design of existing code.*
- Martin Fowler, *Avoid Repetition*, IEEE Software, Jan/Feb 2001 pp.97—99.
- Tom Mens et al. "A survey of software refactoring". TSE 30(2), 2004.
- Y. Yu et al. "Software refactorings guided by softgoals", *REFACE workshop in conjunction with WCRE'03*.
- Y. Yu et al. "Refactor source code into goal models", Technical report.

What's next ...

- A Tutorial on more refactoring practices
 - How to use refactoring in Eclipse?
 - How to use statecharts to represent the refactorings for unstructured code (Webbased software) For example, Squirrel Mail.