Homework 3: Code Reading

Laboratory Date: Thursday, July 2, 2009
Take Home Due: Monday, July 2, 2009

Name : _____

Student Number : _____

Laboratory Time : _____

Instructions for the Laboratory

Objectives

- Open a project from existent source code
- Use Eclipse perspectives and views to facilitate code reading
- Debug a Java program

Grading Checklist (30 points)

By the end of the laboratory session, you need to demonstrate to the TA that you can do the following tasks. You may need to do some preparation in advance to ensure that you can complete all of them in the time available. It's best if you complete all of the tasks and take the quiz on EEE. You can take the quiz as many times as you wish, but only during your laboratory session. You're done when you receive a grade that you are happy with on the quiz, or you run out of time, whichever happens first.

- ☐ Open a project from existing source code
- ☐ Familiarize yourself with the DVD Vendor system
- ☐ Take the quiz on EEE

Task 1: Create a new project from existing source code

Start by unzipping the file provided to you on the webpage.

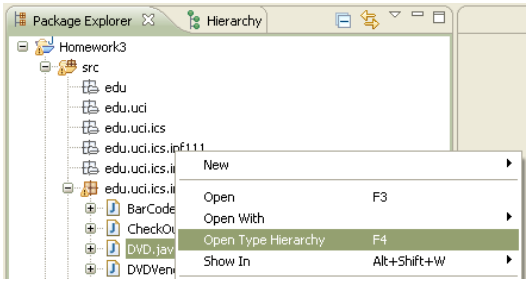
In Eclipse:

a) Create a Java Project. Give it the name Homework3 and select the radio button "create project from existing source". Select the directory of the expanded zip file and click the Finish button.
b) Check that the Project -> Build Automatically menu item is selected.
c) Click on the edu.uci.ics.inf111.dvdvendor.gui.DVDVendorGUI file, which contains the graphical user interface for the DVD Vendor system, and select Run -> Run As… -> Java Application.

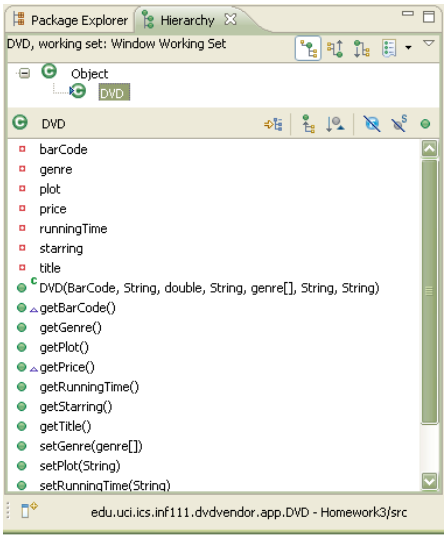
Task 2: Hierarchy View

The Hierarchy View will help us to see the hierarchical relationships among Java elements. In this task, we will examine the hierarchy of the DVD class.

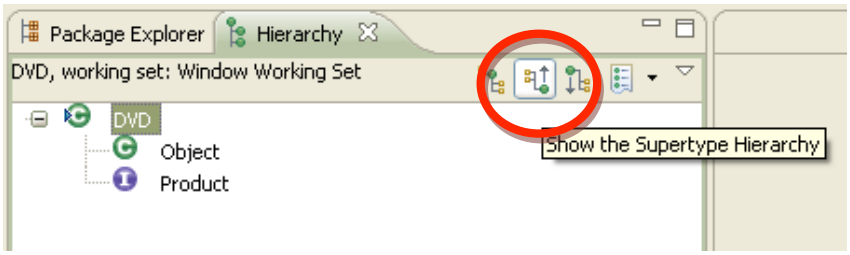
a) In the Package Explorer, right click on edu.uci.ics.inf111.dvdvendor.app.DVD class. Select Open Type Hierarchy. Another option to open the Hierarchy View is to select the java element to be examined and press F4.



b) In the upper part of the Hierarchy View, you will see the hierarchy tree for the examined element. In this case, we see that DVD inherits from Object. In the bottom part, we see the Java elements (attributes, constructors, methods, etc) of the class being examined.



c) Click on the second icon 'Show the Supertype Hierarchy.' It will show the classes and interfaces the examined element is extending or implementing. In this case, we observe that DVD extends the Object class and also implements the Product interface.

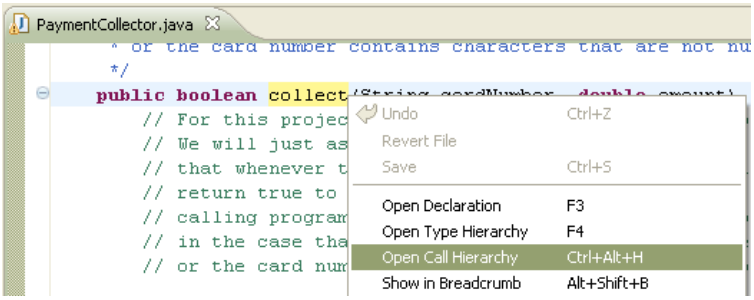


For more information about the Java Browsing Perspective, check: http://help.eclipse.org/stable/index.jsp?topic=/org.eclipse.jdt.doc.user/gettingStarted/qs-Browsing.htm

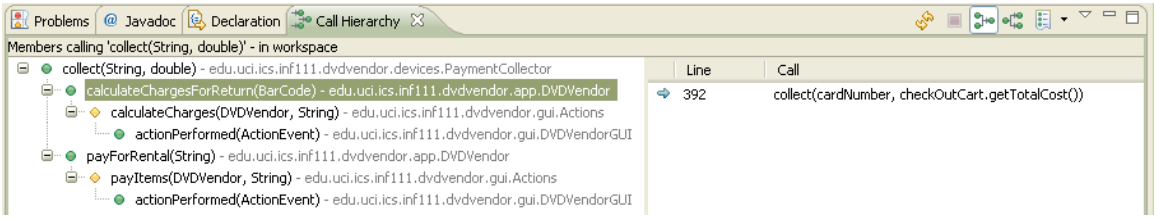
Task 3: Call Hierarchy View

The Call Hierarchy View will help us to identify the Java elements that call a specific method. In this task, we will examine which elements are calling the method collect in edu.uci.ics.inf111.dvdvendor.devices.PaymentCollector.

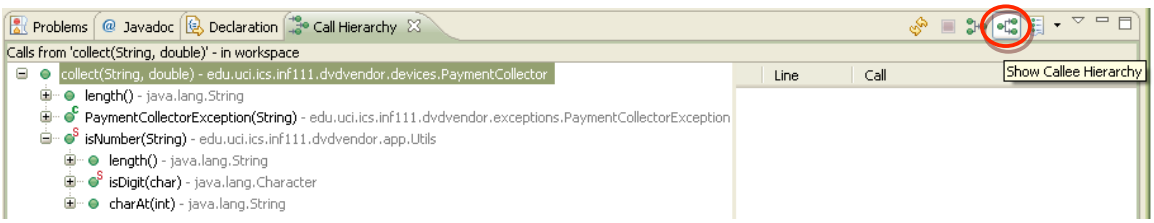
a) In the Package Explorer, double click on the class edu.uci.ics.inf111.dvdvendor.devices.PaymentCollector. In the editor, place your cursor on the collect method. Right-click and choose Open Call Hierarchy. You can also open this view from other views, such as the Outline and others.



b) The Call Hierarchy will show the Java elements that call the specified method. In this case, we observe that two methods are calling collect in PaymentCollector. We can select one of the Java elements and we will see in the right panel the line where this call is made.



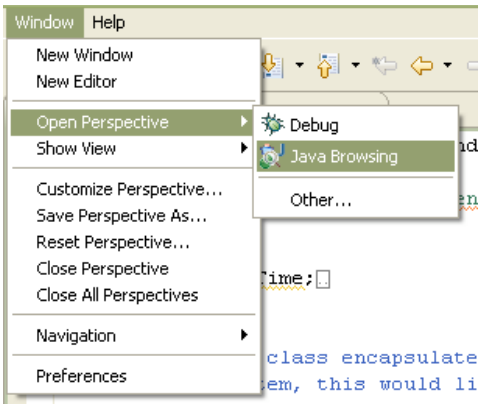
c) Click on the button "Show Callee Hierarchy" and you will see the Java elements that are called from a specific method. In this case, we observe that three methods in three different classes are called from collect in PaymentCollector. We can also expand each of these Java elements to see which methods they call.



Task 4: Java Browsing Perspective

The Java Browsing Perspective will help us to browse and manipulate our code. It shows the same information that the Package Explorer, but with different views and organization. In this task, we will browse the elements of the edu.uci.ics.inf111.dvdvendor.app.Utils class.

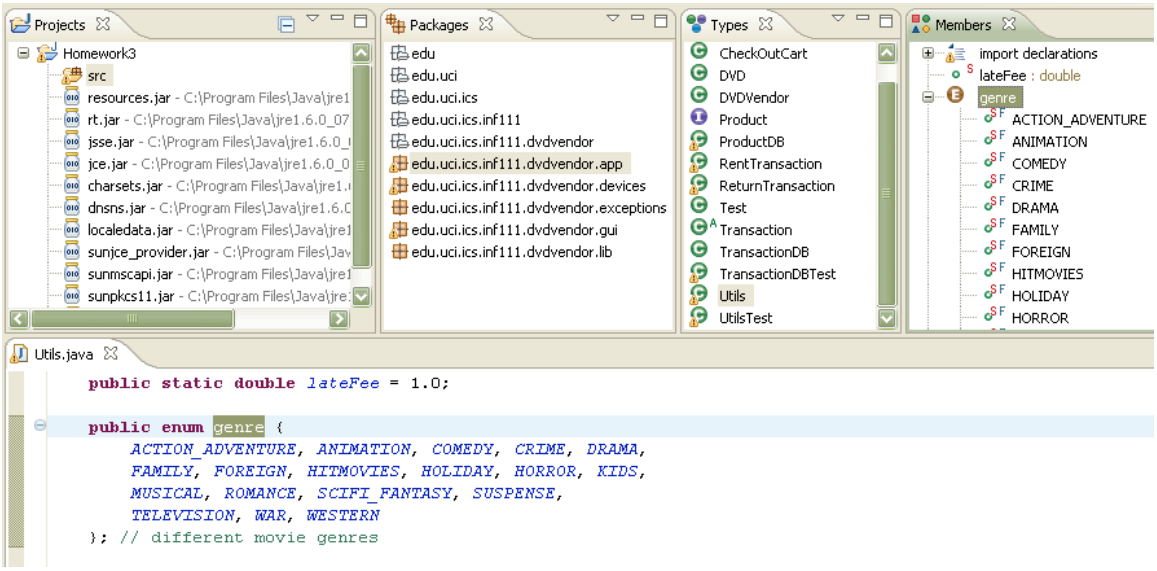
a) Select Window -> Open Perspective -> Java Browsing



On the top part of the window, you will see four views. The editor is on the bottom part. The four views are:

- Projects: it shows the projects that you have in your workspace.
- Packages: it shows the packages for the selected project in the Projects View.
- Types: it shows the different types of java elements you have in the selected package in the Packages View. This view is useful to differentiate between Classes and Interfaces.
- Members: it shows the members of a selected type in the Types View. Members include import declarations, attributes, enumerations, constructors, methods, etc.

b) Select the Homework3 in the Projects View.
Select the edu.uci.ics.inf111.dvdvendor.app package in the Packages View
Select the Utils class in the Types View
Double click on the genre enumeration in the Members View. The editor will show you the details of the selected enumeration.

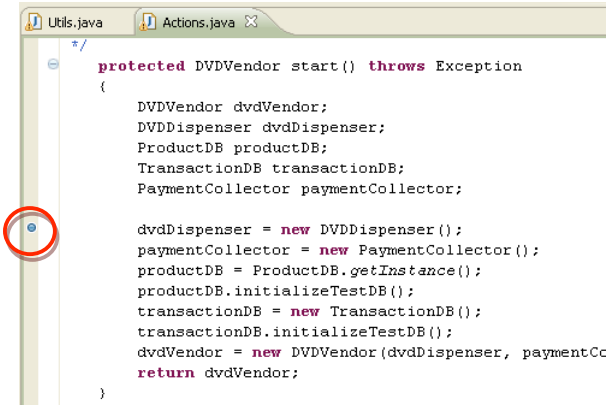


For more information about the Java Browsing Perspective, check: http://help.eclipse.org/stable/index.jsp?topic=/org.eclipse.jdt.doc.user/gettingStarted/qs-Browsing.htm

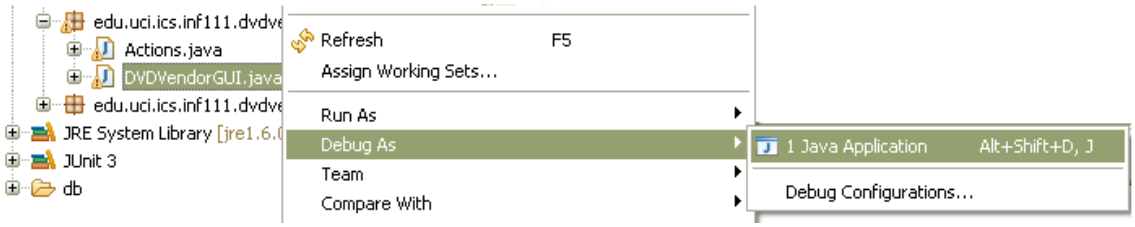
Task 5: Debugging a Java program

Debugging your Java program will help you to run your program step by step, and stop at any particular point to analyze the current values of variables and other elements. In this task, we will debug the method start in the edu.uci.ics.inf111.dvdvendor.gui.Actions class. To do this, we first need to set a break point. A break point is a place where the program will stop during debugging. In our case, it will be in the start method of the Actions class.

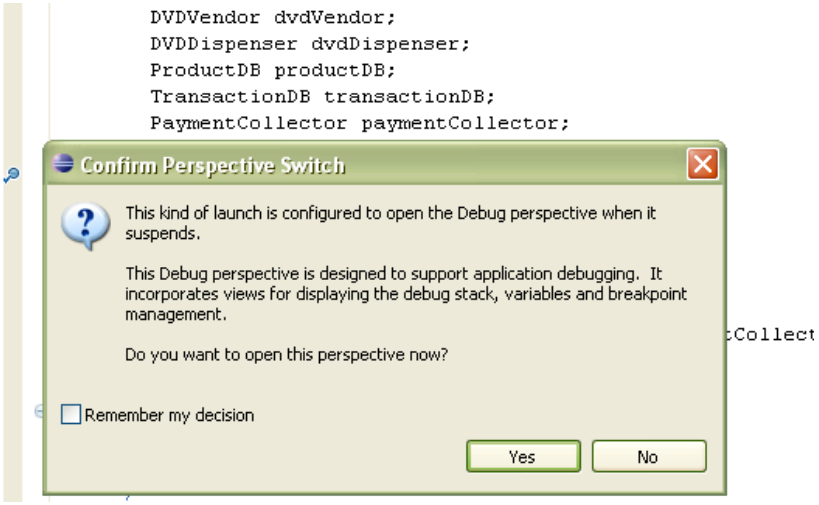
a) Open the Java Perspective. In the Package Explorer, double click on the edu.uci.ics.inf111.dvdvendor.gui.Actions class. To set a break point we need to locate the cursor on the vertical ruler along the left edge of the editor area after the variable declarations in the start method for the Actions class and double click on the ruler to set the breakpoint.



b) In the Package Explorer, select the class edu.uci.ics.inf111.dvdvendor.gui.DVDVendorGUI. Right click on it and select Debug As, and select Java Application.



c) The GUI of the DVD Vendor will appear. Press the Rent DVDs button. A window will popup indicating that the Debug perspective will be opened. Click Yes.



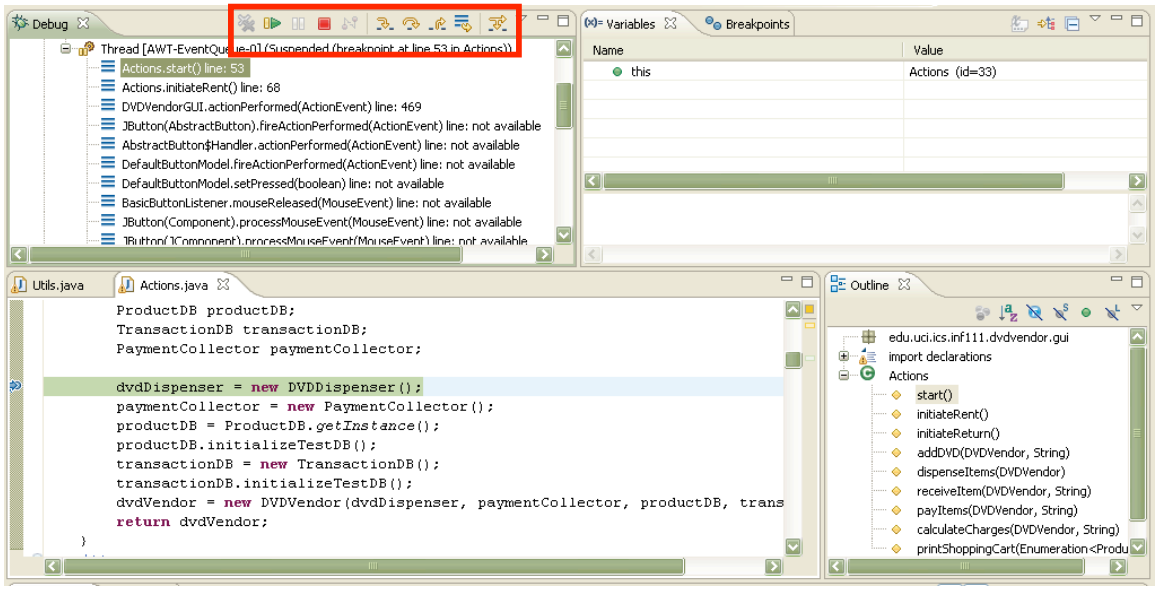
d) The Debug perspective will be opened and it will highlight the line where we set the break point. Some of the actions we can perform are:

Resume : it will allow the program to run until the next breakpoint or until the program is completed.

Terminate : it will terminate the program.

Step into : it will continue the execution at the first line of the method that is invoked in the current line.

Step Over : it will continue the execution at the next line in the same method.



e) Press Step Over two times to reach the method ProductDB.getInstance(). Pay attention to the Variables View that is updating the values of each variable.

f) Press Step Into on the method ProductDB.getInstance() to execute step by step the statements in this method.

g) Play with the different buttons, when you are ready to continue the execution, press the Resume button or terminate the execution with the Terminate button.

For more information about Debugging, check: http://help.eclipse.org/stable/index.jsp?topic=/org.eclipse.jdt.doc.user/gettingStarted/qs-Browsing.htm

Take Home

Based on the code for the DVD Vendor system, answer the following questions:

1. How does the system deal with having more than one DVD with the same title? Include an explanation of how it is handled by the database of products, the Rent transaction, and the Return transaction. (10 points)

2. How does the system figure out which credit card to be charged in the Return transaction? In other words, what is the unique identifier for a transaction? Name one other system that uses a similar approach. (10 points)

3. List all the invoked methods for the following actions:
i) "Add DVD" in the Rent Transaction (20 points)
ii) "Calculate Charges" in the Return Transaction (20 points)