Lab 3: Implementation and Experimental Evaluation of Object-Oriented Programs that Use the Structured Query Language

Assigned: Thursday, January 31, 2013

Due: Thursday, February 14, 2013

Purpose

To familiarize ourselves with some of the fundamental aspects of using the structured query language in an object-oriented program. In particular, to learn more about how to implement and evaluate programs that use Java Objects for SQL (JoSQL). Furthermore, to implement a simple benchmarking framework that you can use to conduct experiments that measure the performance of different JoSQL features. Finally, to perform an empirical study, analyze the results, write up the results in a formal report, and give a group presentation.

Deliverables

In order to satisfactorily complete this laboratory assignment, you must submit the following deliverables. Each of the laboratory notebook entries must be placed inside of the course Web site area that you created. Each of the printed code segments must include a signed pledge on the printed document. The laboratory notebook should be placed inside the course Web site page associated with this laboratory. Make sure that the name of your laboratory notebook is linked to from the Web site page for this laboratory. If you have questions about this issue, please see the Instructor. When you submit this laboratory assignment, please print out the laboratory notebook and use it as the cover sheet for your entire assignment.

Laboratory Notebook, Writing, and Presentation

1. A description of the features that JoSQL provides. You should clearly explain how JoSQL processes a query and then discuss the relative strengths and weaknesses of this approach.

2. A brief description of the steps that you would take to test a program that was written in Java and JoSQL. How would you know if the queries executed by the program were indeed correct? Students may earn extra credit if they provide an implementation of their testing plan.

3. A full report that explains your benchmarking framework and the results that you obtained

from using your benchmarks to evaluate the performance of JoSQL. Your report should include well-written paragraphs, tables of data, graphs, and a statistical analysis of the results. You should submit this deliverable as a stand-alone file instead of placing it in the course Web site.

4. The slides for a short five to eight minute presentation detailing your results from the empirical study. Using technical diagrams, equations, screen shots, short bullet points, tables of data, and graphs, your talk should include the following key points:

    1. A brief summary of your research questions and the benchmarks that each of the group members implemented.
    2. An overview of the empirical results identified by every member of the group, with a commentary on the similarities and differences in the results and a discussion of why these results were evident in your data sets.
    3. A brief commentary on experiments that you could conduct as part of future research in this area.

5. A review of each pitfall or problem that you encountered during the laboratory and a discussion of the steps that you took to overcome this difficulty. If no difficulties were experienced, then please include a record of important commands, notes, etc. that you think will be useful during later laboratories. Your review of these problems should include sufficiently detailed comments so that you can handle the same problem in future laboratory assignments.

Source Code and Output

1. The source code of a complete implementation of the `FileFinder` program. This version of the tool should accept user input and provide a properly formatted output. In particular, your program must accept as input a search root and a file match pattern and then return all of the files under the root whose name match the pattern. Your `FileFinder` program should have detailed comments for every class, each of the methods, and all of the important lines of code.

2. The output from running the `FileFinder` with five different directory roots. For each directory root you should select at least two different file match patterns when you run the program. Furthermore, you should furnish evidence demonstrating that the `FileFinder` has returned the correct file names.

3. A full listing of the source code for your benchmarking framework. You should include a print out of each Java class. Furthermore, your classes should contain detailed documentation that explains the meaning and purpose of each benchmark. Your benchmarking program should also have detailed comments for every class, each of the methods, and all of the important lines of code.

4. All of the unprocessed data files associated with your empirical study. If you collect your data in separate files, you should explain the naming convention of each file and then submit a printed version of every file.

All of your Java code and output must be printed with a command that ensures that you have one or two columns of output and a descriptive header at the top of your page. In particular, the descriptive header should include the date on which you printed your program.

Installing and Using JoSQL

Download and install the programming tool called JoSQL. You can download the binary version of JoSQL and then install the program by placing two JAR files into your `CLASSPATH` environment variable, as specified in JoSQL's README file. Next, please download and compile the `FileFinder.java`. At this point, you should edit the source code of the `FileFinder` so that it will search a specific directory in your home account. Furthermore, you may want to change the file match pattern to something instead of LIKE '%ja%'. Can you get the `FileFinder` program to work properly? Please see the instructor if you are not able compile and execute this application.

After editing and compiling the `FileFinder` you should extend it with several additional features. First, you should add two command line arguments: the name of the root directory and the chosen match pattern. Your enhanced version of `FileFinder` should use the first input parameter when it populates the array list with all of the files under the root directory. Next, the enhanced `FileFinder` must use the second input parameter to run the user-specified SQL `select` statement. Finally, you should run your program with a wide variety of inputs in order to demonstrate that it is working properly. As you complete this part of the laboratory assignment, make sure that you turn in all of the requested source code and output.

Empirical Analysis of Query Languages in Object-Oriented Programs

The existence of JoSQL makes it possible to frame and answer many interesting questions about the performance associated with using the structured query language in an object-oriented program. For instance, would the performance of `FileFinder` vary if you used a `LinkedList` or a `Vector` instead of an `ArrayList`? JoSQL includes many different types of SQL statements -- what is the performance of different querying constructs? Moreover, it would also be interesting to compare the performance of JoSQL to a hand-coded alternative that uses iteration constructs to directly search a data structure for matching objects.

As the final part of this laboratory assignment, you are responsible for implementing a simple benchmarking framework for JoSQL. Your framework should contain, at minimum, two benchmarks that can be run in different configurations. You should design your benchmarks so that they will enable you to answer specific questions about the performance trade-offs associated with using the structured query language for Java objects. After completing the implementation of your benchmarks, you should run them to collect performance results concerning time and (if possible) space overheads. Please carefully execute the benchmarks multiple times and record the empirical results in a structured fashion.

In order to analyze the results, you should construct data tables and visualizations. For instance, you may want to write programs in the R language for statistical computation. After developing an understanding of the fundamental trends in your data sets, you should write a report that details your findings. The report should explain your benchmarks, the evaluation metrics, and the key empirical results. In particular, you should observe the performance trends and explain why these trends are evident in your data set. Whenever possible, you should comment on the practical implications of your results for software developers who may use JoSQL.

You are responsible for working in a group to coordinate the experiments that you conduct. In particular, you should make sure that each member of the group implements different benchmarks and then conducts experiments in a uniform manner. After the members of the group have individually completed the experiments and recorded and analyzed their results, you should meet together to discuss the trends that you found in the data sets. While students are responsible for writing their own reports that focus on their own data sets and analyses, they should work together in a group to prepare a single presentation for the laboratory session on which this assignment is due. After explaining the purpose and configuration of your experimental study, the presentation should highlight the similarities and differences in the results developed by each group member. Finally, you should draw some conclusions about which algorithmic approach and/or data management technique is the fastest. Please note that each group member is responsible for independently conducting their own experiments, writing their own report, and adequately participating in the completion of the slides and the final presentation.
