

Three small projects for an „Advanced Programming in Java” course

Ioan Jurca
„Politehnica” University of Timisoara
Romania

Agenda

- Course Syllabus
- Types of homework
- Project 1: Concurrent programming
- Project 2: A tool for DB manipulation
- Project 3: RMI application: a chat system
- Experience with the projects
- Conclusions

Course Syllabus

- The course is a level 4 course in a Computer Science Program at an American college for adult students (CMSC 480)
- It covers the following subjects:
 - Multithreading (2 weeks)
 - Collections (1 week)
 - Networking (2 weeks)
 - JDBC (1 week)
 - RMI (1 week)
 - Web Services (1 week)
 - Security (2 weeks)
 - Develop capstone project (3 weeks)
- Textbook for the course: C. Horstman, G. Cornell: *Core Java*, vol. 2 (8th edition) – does not cover multithreading and collections

Types of homework

- Homework account for 60% of the final course grade
- 10% of the grade is given by online participation
- 30% of the grade is given by „authentic assessment”, i.e. development of a larger „capstone” project – specified as a Web online store application (there is no final exam)
- Two types of homework:
 - projects (45%)
 - essay (15%)

Project 1: Concurrent Programming

- Classical problem of producer/consumer with bounded buffer, with the following specific requirements:
 - any number of producers and consumers
 - size of buffer specified by the user
 - a thread produces or consumes an element in a time computed as `Math.random() * 10000`
 - the output shows all events in the system
 - at least 3 solutions must be submitted, using the following synchronization mechanisms:
 - synchronized methods, with `wait()` and `notifyAll()`
 - locks and conditions (Java 5)
 - semaphores (Java 5)
 - blocking queues (java 5)

Project 1 output

```
run:
Welcome to the multithreading application.
Would you like to specify the number of producers,
consumers and buffer capacity?
(Enter 'y' for YES and 'n' for NO.
y
Please enter the number of producers.
5
Please enter the number of consumers.
5
Please enter the desired buffer capacity.
10
Ok. Starting program with 5 producers, 5 consumers, and a buffer
size of 10.
Press CTRL+C to end program.
Producer 5 adding item 5000
Remaining buffer capacity: 9
Consumer 5 got item 5000
Producer 3 adding item 3000
Remaining buffer capacity: 9
```

Project 2: A tool for DB manipulation

Main requirements:

- starts from a tool given in the course textbook which uses database metadata to display and browse DB tables
- must provide facilities for adding new rows in a table and for adding new tables to the database
- uses the Derby database which is integrated with the NetBeans IDE
- application must work outside the IDE
- user manual, including installation instructions and screenshots for different functionality

Project 2: User manual – a paragraph

Using ViewDB

Before using ViewDB, you must review and make any necessary changes to the file `database.properties`. This file is similar to `ij.properties` above in that it defines the database connection information, however the property names are slightly different.

You should also edit the file `ViewDB.bat` and change the variable on the first line, "`DERBY=...`", to point to the root directory for your JavaDB install.

To start ViewDB simply enter "`ViewDB`" at the command prompt to run the bat file.

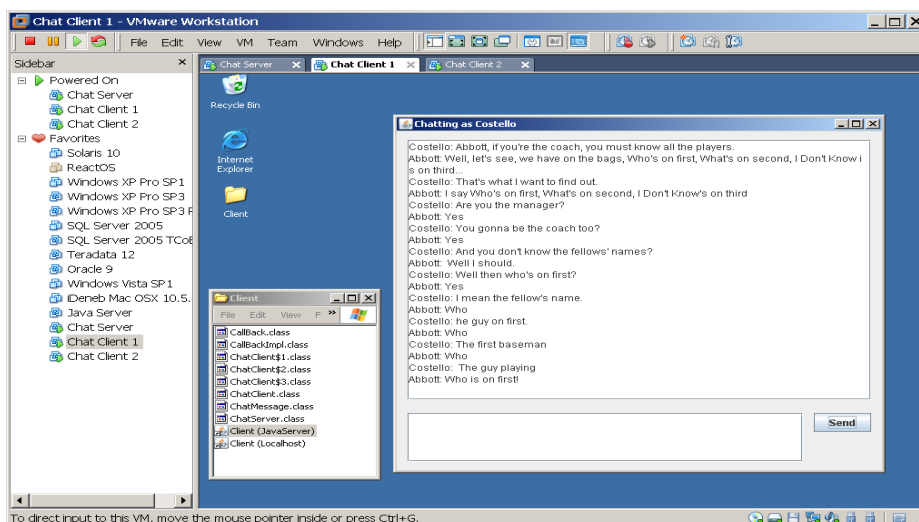
The following screen shots are based on the sample data provided.

Project 3: RMI Application – a chat system

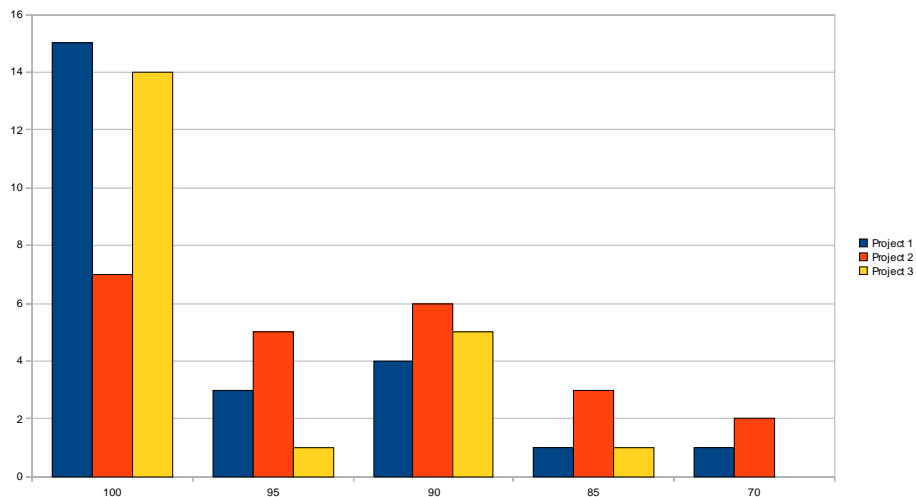
Main requirements:

- The chat server keeps track of users, requiring their name and host
- Any number of users can ask for connection, send messages, see messages from all other connected users and can disconnect
- The solution must be implemented using RMI for communication between users and server, and the *Observer* design pattern ('callbacks')
- user manual must be provided

Project 3: A screenshot



Results



Conclusions

- the 3 projects are quite manageable for a one-semester course and deal with essential techniques offered by the Java platform
- students encountered most difficulties with the DB project, especially in configuring correctly the DB access
- another source of difficulty: testing RMI in a 'true' multi-machine environment
- an interesting range of solutions for same requirements from the 20+ students