The Craftsman: 17
SMCRemote Part VII
Call the Guards

Robert C. Martin
17 Aug, 2003

...Continued from last month. You can download last month's code from:

www.objectmentor.com/resources/articles/CraftsmanCode/Craftsman_16_SMCRemote_VI_Wham_Bam.zip

Dear Diary, my first week as an apprentice for Mr. C. is over. I've learned a lot in that week, and I've also made a real mess of things. Here's a recap: On Monday Jerry had me write a program to generate prime numbers. On Tuesday he had me write a program to generate prime factors. Wednesday was spent getting the SocketServer working. Thursday we started working on SMCRemoteClient, and I Micahed Jerry. I met Jasmine that afternoon in the Journeymen's lounge. Friday was the most embarrassing day of my life. I haven't seen Jasmine (Ms J) since we finished working Friday afternoon. What a roller coaster of a week. (Diary, just what is a roller coaster anyway?)

I spent the weekend pretty depressed. I was sure that I was going to be transferred to sanitation & recycling, reactor cleanup, or mildew scrubbing. I barely bothered to get dressed at eat.

Anyway, now it's Monday evening, and I'm writing to tell you about the first day of my second week. It didn't turn out badly at all. In fact, I'm starting to feel optimistic again.

I woke up on Monday full of dread. After breakfast I shambled over to the workroom expecting to see Ms. J waiting for me at my workstation. Instead I saw a plumpish middle-aged woman with shoulder length blonde hair, and a grandmother smile that spoke of younger years. When she looked at me her eyes twinkled and she gave me a wide grin. "You must be Alphonse." She said, holding out her hand. "My name is Jean. I've heard a great deal about you. Are you hungry? Young men like you are always hungry. I've got a nice sandwich here in my basket if you'd like it, or perhaps you'd like an apple or a banana.

There was, indeed, a basket on the floor next to her. It looked like it contained quite a bit more than just sandwiches and bananas. I shook her hand and said: "Yes Ma'am -- I mean, no Ma'am -- I mean -- Yes I am Alphonse, and no thank you for the sandwich, and I'm -- er -- pleased to meet you."

She looked at me sternly, yet with a big motherly grin. "Now we'll have none of that Ma'am nonsense. I'm certainly not anybody's Ma'am. The very idea! You just call me Jean, dear."

"Uh, well, thank you Jean. Can you tell me where Ms. J is?"

"Who dear?"

"er -- Jasmine. She and I are supposed to be working together."

"Goodness me, I've never heard her called Ms. J before. Did she ask you to call her that? I don't think anyone else calls her anything but Jasmine. Such a lovely girl, isn't she? Anyway, dear, Mr. C. asked her to work on a different project for the time being, and so I'll be working with you from now on."

"You?" I was caught completely unprepared for this. "Oh." I said intelligently. "Uh..."

"Now you sit right down here, dear, and let me tell you what I've been thinking."

"Thinking?"

"Yes dear! I've been looking over this program of yours for the last half-hour -- I like to get to work early you know -- not that you need to get here any earlier than usual -- I know a growing boy needs his sleep and breakfast. Anyway I've been quite pleased with this program. You've got a very pleasant suite of tests, and the code is quite easy to read and, really quite nicely structured, I'd say. But there's one thing that puzzles me dear."

"Uh -- puzzles?"

"Yes, dear! I've been looking all through this program and I can't find the main function. When were you going to write it dear?"

"Uh, well, Jerry said..."

"Oh let me just guess what Jerry said. Jerry's a nice lad, dear, but sometimes I think he could use a few more clues about things. But I shouldn't be saying anything bad about anyone. Jerry is a fine programmer, dear, and you just never mind about what I said. Now, let's write that main function. Would you like to start? My fingers are feeling a bit stiff this morning. Take my advice and don't get old, dear."

Dazed by the sheer inundation of words I took the keyboard and began typing:

```
public void main(
```

"Oh, now, dear! How long have you been working here? You've got to write a test first, dear. You can't just go off writing the main function! Where would we be if everyone just wrote the functions without writing the tests first? I can tell you where -- In a pickle, that's where.   No, dear, you delete that and write a test first.

She took a pair of knitting needles out of her basket and began working on a project of some kind. It looked vaguely like a shawl. She started softly humming a simple tune while I deleted the characters I had written and started over.

How do you test the main function? I guess the best answer is simply to call it and then make sure it did what it was supposed to do. The main function interprets the command line options, so calling it is just a matter of passing in the right options. So I started typing again:

```
public void testMain() throws Exception {
  SMCRemoteClient.main(new String[]{"myFile.sm"});
```

Jean looked up from her knitting and said: "That's nice dear, but where does myFile.sm come from? We can't just have that laying around, can we? No, we'll have to create it right here, won't we dear? And don't forget to delete it when you are done, dear. Nothing worse than a bunch of old files hanging around, I always say."

So I continued typing:

```
public void testMain() throws Exception {
  File f = createTestFile("myFile.sm", "the content");
  SMCRemoteClient.main(new String[]{"myFile.sm"});
  f.delete();
  File resultFile = new File("resultFile.java");
  assertTrue(resultFile.exists());
  resultFile.delete();
}
```

Jean had finished another row of her shawl while I typed. She looked up just as I finished and said: "Now, dear, that's just fine, but I do think that main might not have enough time to finish executing before that delete takes effect. Remember, dear, you've got all those socket threads running, and one of them could easily still be going by the time main returns. No, dear, don't do anything about it now; just keep it in mind. Now I think you'd better write main, don't you?"

I thought to myself that this was a pretty sharp old lady, and I started writing the main function.

```java
public static void main(String[] args) {
  SMCRemoteClient client = new SMCRemoteClient();
  client.setFilename(args[0]);
  if (client.prepareFile())
    if (client.connect())
      if (client.compileFile())
        client.close();
      else {  // compileFile
        System.out.println("failed to compile");
      }
    else { // connect
      System.out.println("failed to connect");
    }
  else { // prepareFile
    System.out.println("failed to prepare");
  }
}
```

"Oh my, now nice that looks! I think it's very clever how you commented those `else` statements. Though I wonder if it wouldn't be more readable if you inverted the sense of the `if` statements and used them as guards. No, dear, don't change it yet. Let's see if it works first. It's never worth making lots of changes until you know whether the program work or not, don't you agree? First make it work, *then* make it right."

So I ran the test, and it worked first time.

"Oh, that's just splendid!" She said while glancing up from her knitting. "Now, let's change that `if` statement."

So I changed the function so that the `if` statements were guards.

```java
public static void main(String[] args) {
  SMCRemoteClient client = new SMCRemoteClient();
  client.setFilename(args[0]);
  if (!client.prepareFile()) {
    System.out.println("failed to prepare");
    return;
  }
  if (!client.connect()) {
    System.out.println("failed to connect");
    return;
  }
  if (!client.compileFile()) {
    System.out.println("failed to compile");
    client.close();
    return;
  }
  client.close();
}
```

Jean put her knitting down into her basket and looked carefully at the code. "Well I do think that looks a little better, though I can't say I care for that duplicated close statement. And the violation of single/entry, single/exit is a bit bothersome. Still, it's better than all that indentation, don't you agree? Of course we could change those three functions to throw exceptions, but then we'd have to catch them, and that would be a bother. No, let's leave it like that for the time being. Now, I think it's time for a break, don't you? Would you like to carry my basket to the break room for me? I'm always over-packing it and it gets so heavy after awhile you know."

To be continued...

The code that Alphonse and Jasmine finished can be retrieved from:
www.objectmentor.com/resources/articles/CraftsmanCode/ Craftsman_17_Call_the_Guards.zip